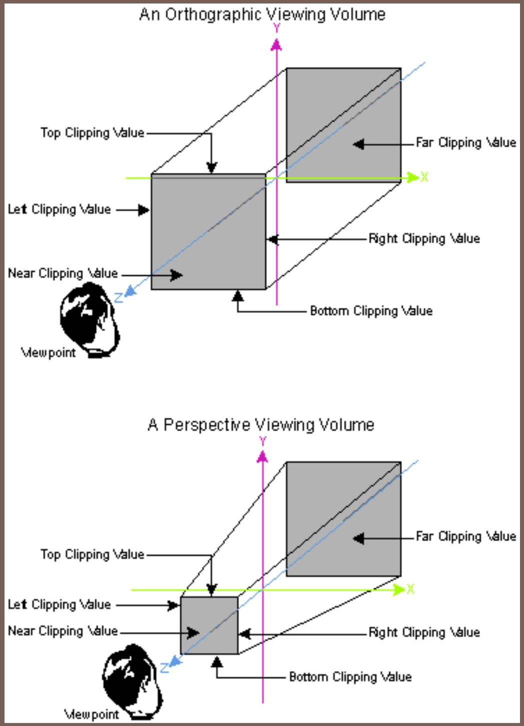


COMPUTER GRAPHICS

Helmuth Trefftz – EAFIT
Traducida por Juan G. Cárcamo - EAFIT

Agenda

- Traslación
- Escalamiento
- Rotaciones
  - Al rededor del eje X
  - Al rededor del eje Y
  - Al rededor del eje Z
- Perspectiva

Transformaciones 3D

Traslaciones:

$$T(d_x, d_y, d_z) = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformaciones 3D

Escalamiento:

$$S(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformaciones 3D

Escalamiento:

Al igual que en 2D, la transformación de escalamiento puede tambien mover el objeto, si este no se encuentra en el origen del plano cartesiano.

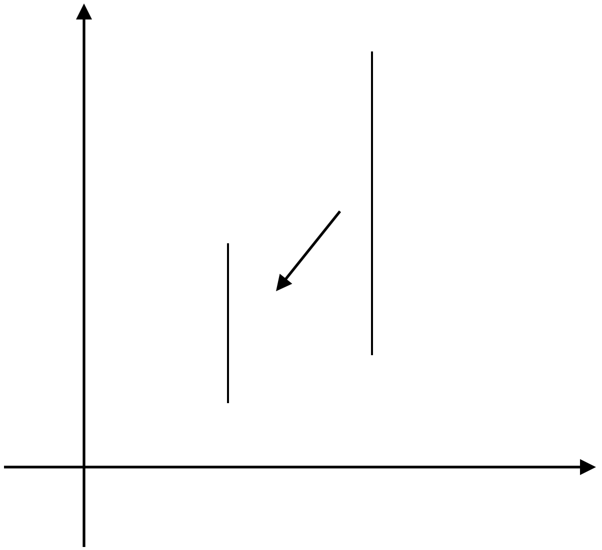
Esto se puede prevenir, trasladando el objeto al origen, escalandolo y trasladandolo nuevamente al punto original.

Caso 2D

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

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

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



Combinar las trasnformaciones 3D

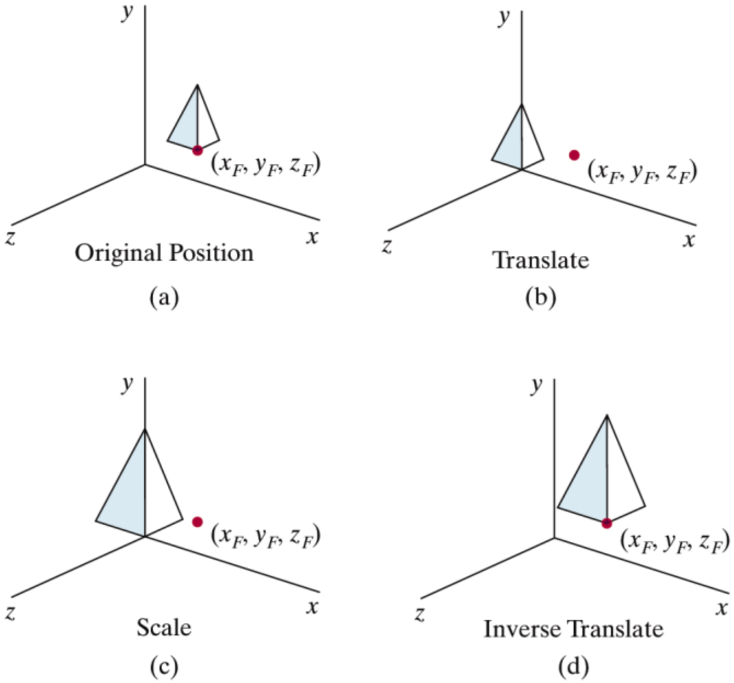


Figure 5-51

A sequence of transformations for scaling an object relative to a selected fixed point, using Eq. 5-110.

Rotaciones en 3D

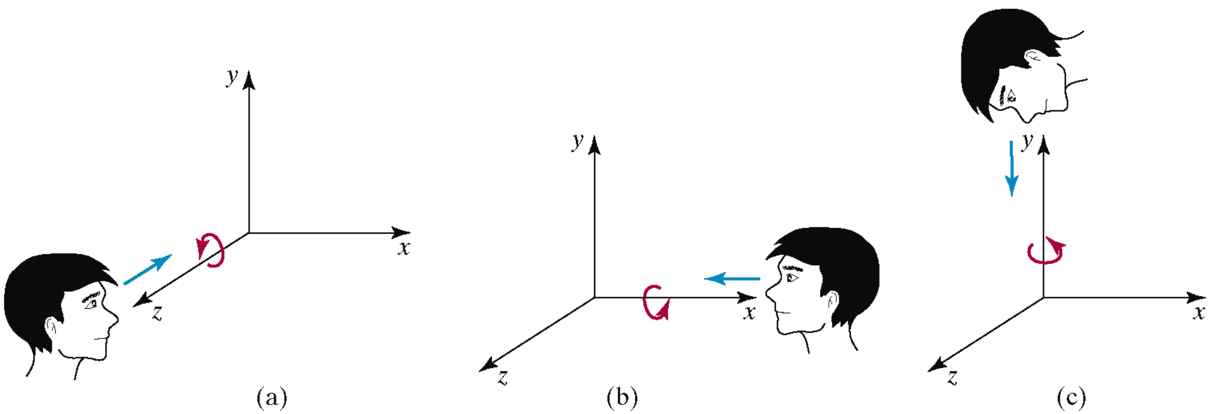
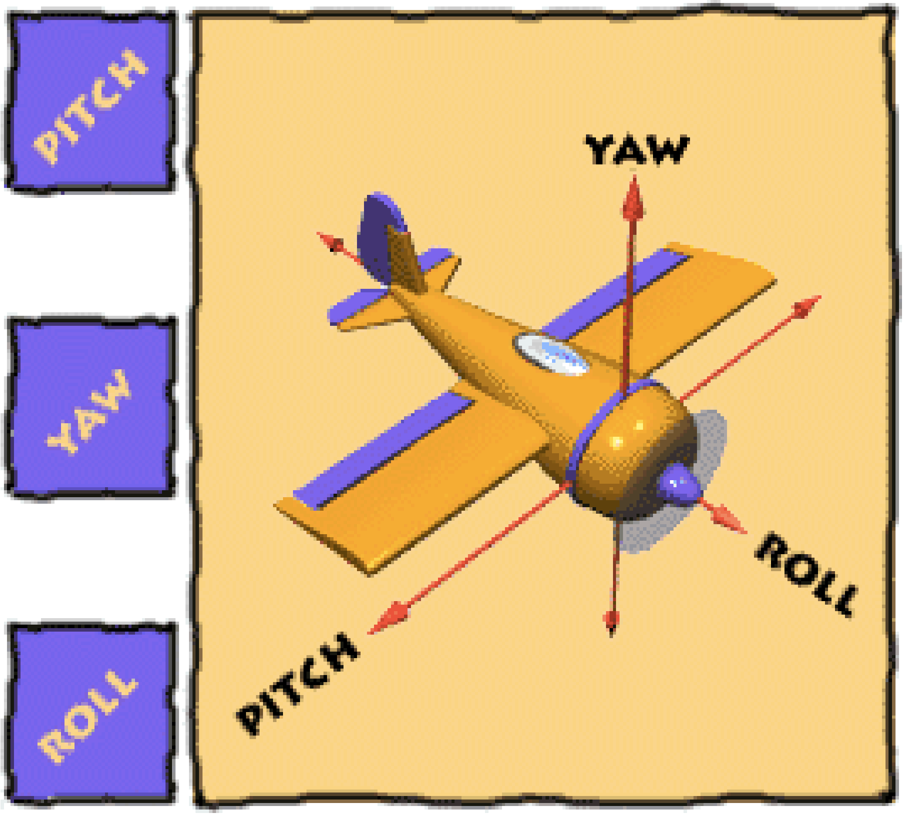


Figure 5-36

Positive rotations about a coordinate axis are counterclockwise, when looking along the positive half of the axis toward the origin.

Rotaciones en 3D



- Image Source: http://www.vrarchitect.net/anu/cg/Transformation/printNotes.en.html

3D Transforms

Usamos la regla de la mano derecha para determinar el eje de rotación.

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

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

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

Transformaciones en 3D

Rotación:

Al igual que en 2D, los objetos rotan al rededor del origen.

Para evitar traslaciones no deseadas, trasladamos primero el objeto al origen, es rotado y finalmente lo trasladamos de vuelta a la posición original.

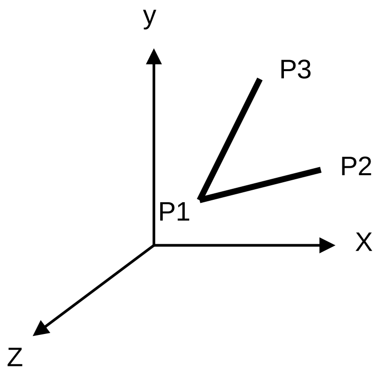
Transformadas inversas

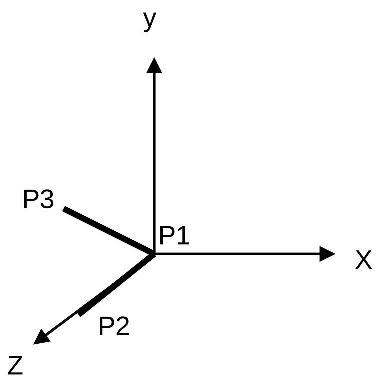
- Inversa de T: cambiar los signos de dx, dy, dz
- Inversa de S: cambiar sx, sy, sz por su reciproco.
- Inversa de R: Cambiar el signo del angulo

Combinar las rotaciones 3D

Como hacer que un punto rote en 3D al rededor de un eje arbitrario?

Rotación al rededor de un eje arbitrario





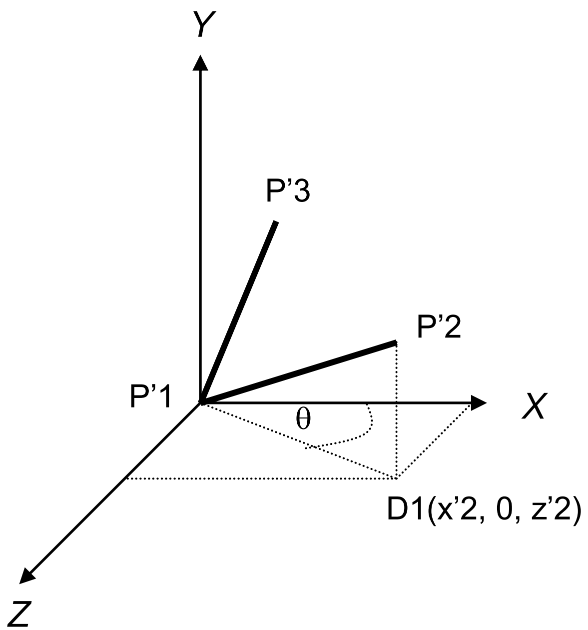
- P3 rotará al rededor del eje definido por P1P2
- (i) Realice una serie de traslaciones y rotaionaes hasta que el eje P1P2 coincida con el eje z.
- (ii) Rote al rededor de *z*.
- (iii) Aplique la inversa de todas las transformaciones realizadas en (i)

Paso 1

- Traslade P1 al origen:

$$T(-x1,-y1,-z1)=\begin{bmatrix}1 & 0 & 0 & -x1\\ 0 & 1 & 0 & -y1\\ 0 & 0 & 1 & -z1\\ 0 & 0 & 0 & 1\end{bmatrix}$$

Step 2



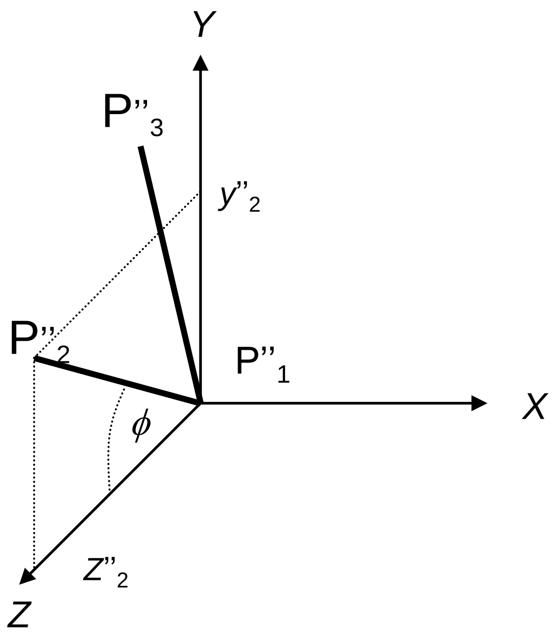
- Rote en el eje *y para que* P'1P'2 esté en el plano YZ
- Angle: -(90-θ) = θ – 90

$$\cos(\theta - 90) = \sin\theta = \frac{z'_2}{D_1} = \frac{z_2 - z_1}{D_1}$$

$$\sin(\theta - 90) = -\cos\theta = \frac{x'_2}{D_1} = \frac{x_2 - x_1}{D_1}$$

$$D_1 = \sqrt{(z_2 - z_1)^2 + (x_2 - x_1)^2}$$

Step 3



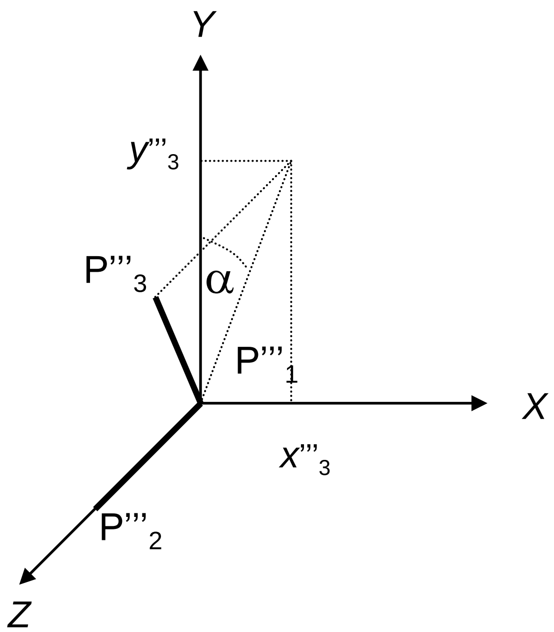
- Rote al rededor del eje *x* para que P"1P"2 coincida con el eje *Z* (note que P"1P"3 no necesariamente estan en el plano YZ):

$$\cos\phi = \frac{z''_2}{D'_2}$$

$$\sin\phi = \frac{y''_2}{D'_2}$$

$$D_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Step 4



- Rote en el eje z el Angulo deseado:

$$P_3''' = \begin{bmatrix} x_3''' & y_3''' & z_3''' \end{bmatrix}^T =$$

$$R_x(\phi)\cdot R_y(\theta - 90)\cdot T(-x_1 \quad -y_1 \quad -z_1)\cdot P_3$$

$$\cos\alpha = \frac{y'''_3}{D_3}$$

$$\sin\alpha = \frac{x'''_3}{D_3}$$

$$D_3 = \sqrt{x_3'''^2 + y_3'''^2}$$

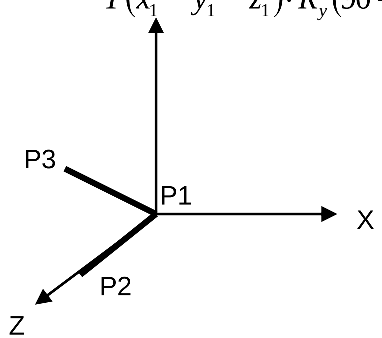
Final transform:

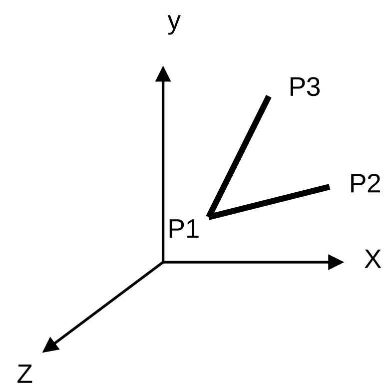
- Complete la transformación

$$R_z(\alpha)\cdot R_x(\phi)\cdot R_y(\theta-90)\cdot T(-x_1 \quad -y_1 \quad -z_1)$$

- Deshaga la primera transformación:

$$T(x_1 \quad y_1 \quad z_1)\cdot R_y(90-\theta)\cdot R_x(-\phi)$$





Other 3D Transformations

- What does the following transformation do?

$$M_{reflect} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- It converts from a right-hand system into a left-hand and vice versa

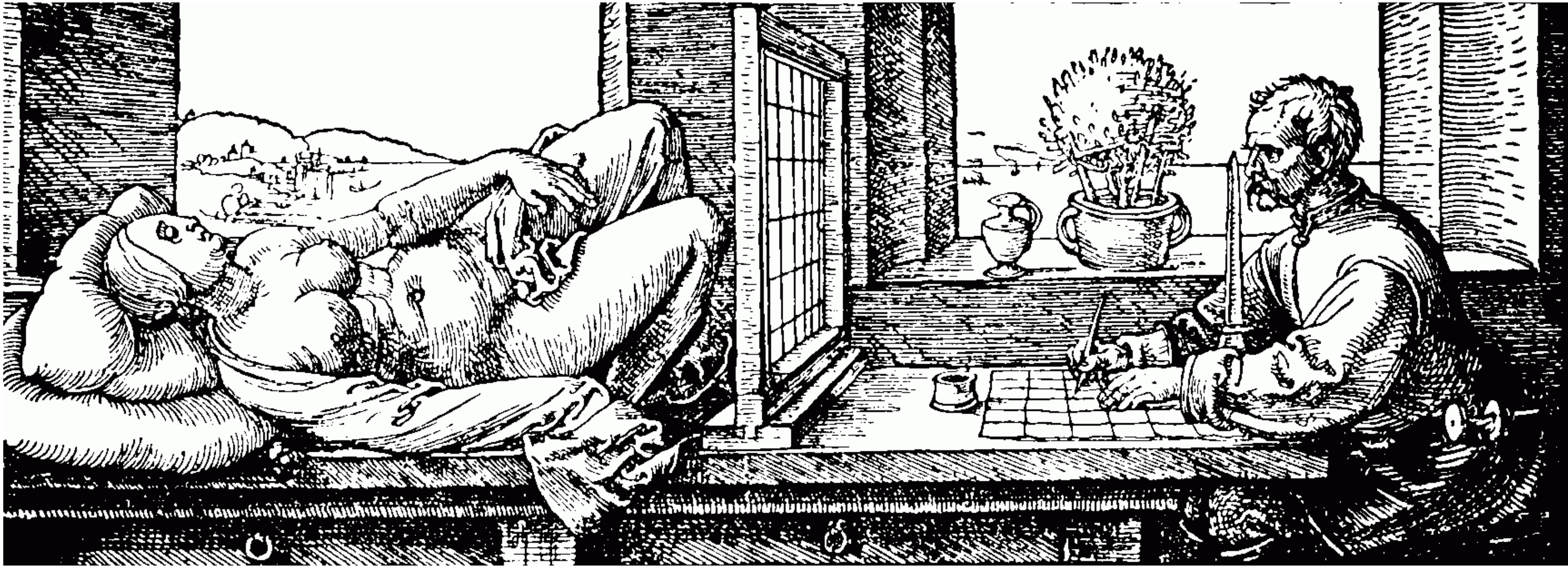
3D Shears

- What does the following transformation do?

$$M_{shear} = \begin{bmatrix} 1 & 0 & sh_{zx} & 0 \\ 0 & 1 & sh_{zy} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

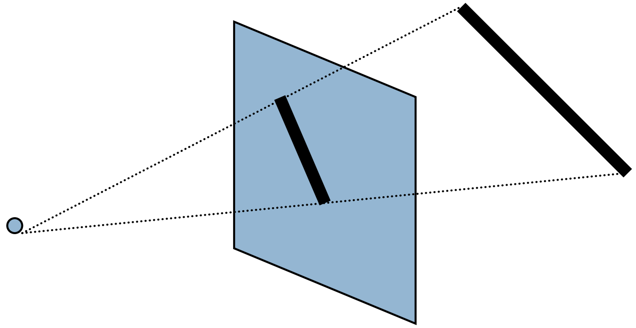
- Shears the x and y coordinates as z increases.

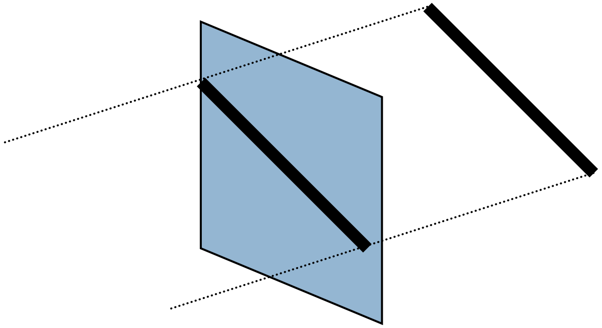
Projection Transformations



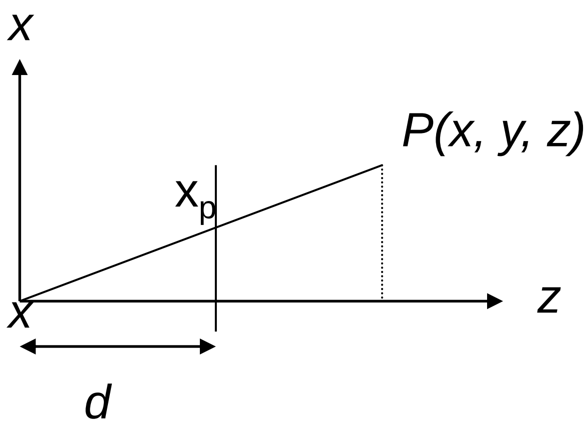
Projection Transformations

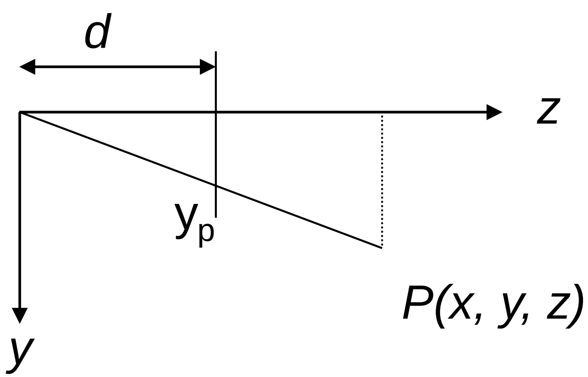
- Projections: *project* a 3D object into a flat, 2D surface
- Types of projection:
  - Perspective
  - Parallel





Perspective





- Projection plane: Orthogonal to the z axis, distance *d* from the origin.

$$\frac{x_p}{d} = \frac{x}{z}; \frac{y_p}{d} = \frac{y}{z}$$

$$x_p = \frac{d \cdot x}{z} = \frac{x}{z/d}; y_p = \frac{d \cdot y}{z} = \frac{y}{z/d}$$

Perspective

- As matrix multiplication:

$$M_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix}$$

$$\begin{bmatrix} X & Y & Z & W \end{bmatrix}^T = \begin{bmatrix} x & y & z & z/d \end{bmatrix}^T$$

- Dividing by W:

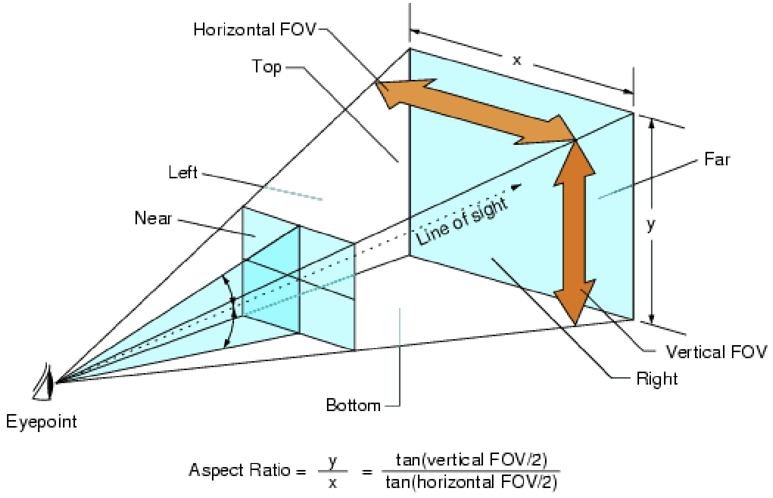
$$\begin{pmatrix} x_p & y_p & z_p \end{pmatrix} = \begin{pmatrix} \dfrac{x}{z/d} & \dfrac{y}{z/d} & d \end{pmatrix}$$

Perspective – For programming purposes

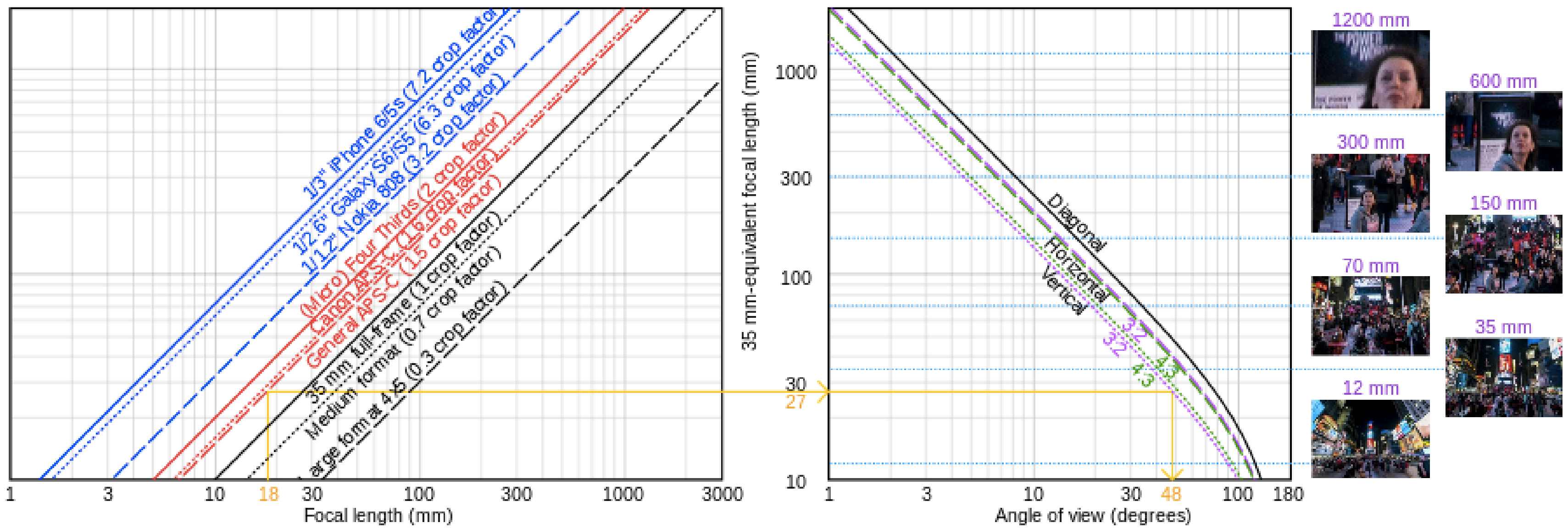
- If the observer is at the origin, looking in the +Z direction, the X axis grows to the left of the viewport window.
- It is simpler to put the observer in the origin, looking in the –Z direction.
- In this case, put the object to draw along the –Z axis, and d becomes negative.
- In this situation, X grows to the right of the viewport window.

What FOV is normal?

- The field of view angle corresponds to normal lenses?
- Consoles: 60 degrees
- PC: 90 to 100 degrees
- Source: (2) and (3)



What FOV is “normal”?



Log-log graphs of focal length vs crop factor vs diagonal, horizontal and vertical angles of view for film or sensors of 3:2 and 4:3 aspect ratios. The yellow line shows an example where 18 mm on 3:2 APS-C is equivalent to 27 mm and yields a vertical angle of 48 degrees.

What FOV is "normal"?

- From Wikipedia: Angle of View:

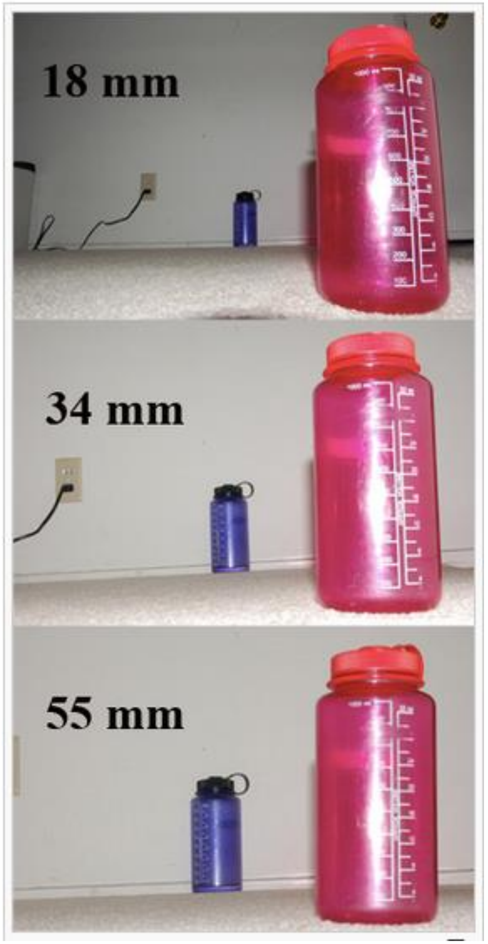
Example [edit]

Consider a 35 mm camera with a lens having a focal length of $F = 50$ mm. The dimensions of the 35 mm image format are 24 mm (vertically) × 36 mm (horizontal), giving a diagonal of about 43.3 mm.

At infinity focus, $f = F$, and the angles of view are:

- horizontally, $\alpha_h = 2\arctan\frac{h}{2f} = 2\arctan\frac{36}{2\times 50} \approx 39.6^\circ$
- vertically, $\alpha_v = 2\arctan\frac{v}{2f} = 2\arctan\frac{24}{2\times 50} \approx 27.0^\circ$
- diagonally, $\alpha_d = 2\arctan\frac{d}{2f} = 2\arctan\frac{43.3}{2\times 50} \approx 46.8^\circ$

How Focal Lenth affects perspective



How lens choice affects FOV

28 mm lens, 65.5° × 46.4°

50 mm lens, 39.6° × 27.0°

70 mm lens, 28.9° × 19.5°

210 mm lens, 9.8° × 6.5°

Bibliography

- (1) Reading: Hearn & Baker. Sections 5-9 to 5-16.

- (2) http://en.wikipedia.org/wiki/Field_of_view_in_video_games
- (3) http://en.wikipedia.org/wiki/Angle_of_view#Focal_length

Credits

- FOV image:
  - http://www.incgamers.com/2013/05/why-good-fov-options-are-crucial-to-pc-games

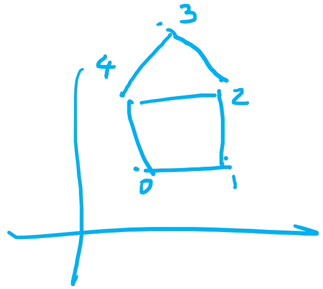
2D

5

100 100
200 100
200 200
150 250
100 200

6

0 1
1 2
2 3
3 4
4 0
2 4



clase

· Dib casita

d = -1000

3D

$$x_p = \frac{x}{z/d} = \frac{100}{-1100/-1000} = 90$$

9 x y z

0) 100 100 -1000
1) 200 100 -1000
2) 200 200 -1000
3) 150 250 -1050
4) 100 200 -1000

5) 100 100 -1100
200 100 -1100
200 200 -1100
100 200 -1100

16

0 1
1 2
2 4
4 0

5 6
6 7
7 8
8 5

0 5
1 6
2 7
4 8

2 3
7 3
8 3
4 3

Transf → 150, 150, -1050