Geometría Proyectiva

15 de Octubre, 2010

Informática Gráfica

Ingeniería Superior Informática

Coordenadas Homogéneas

- Las coordenadas homogéneas nos permiten representar de la misma manera puntos y vectores en el espacio afín:

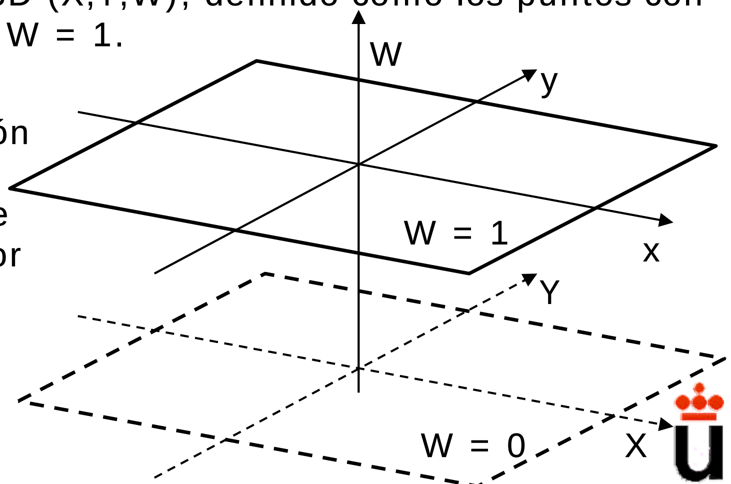
$$\mathbf{p} = p_x\vec{\mathbf{i}} + p_y\vec{\mathbf{j}} + p_z\vec{\mathbf{k}} + 1 \cdot \mathbf{o} = (p_x, p_y, p_z, 1)$$

$$\vec{\mathbf{v}} = v_x\vec{\mathbf{i}} + v_y\vec{\mathbf{j}} + v_z\vec{\mathbf{k}} + 0 \cdot \mathbf{o} = (v_x, v_y, v_z, 0)$$

- Los vectores están asociados a direcciones

Espacios Afín y Proyectivo

- El espacio afín 2D (x,y) es un subespacio del espacio proyectivo 3D (X,Y,W), definido como los puntos con coordenada W = 1.
- La proyección al espacio afín consiste en dividir por la coord. W.

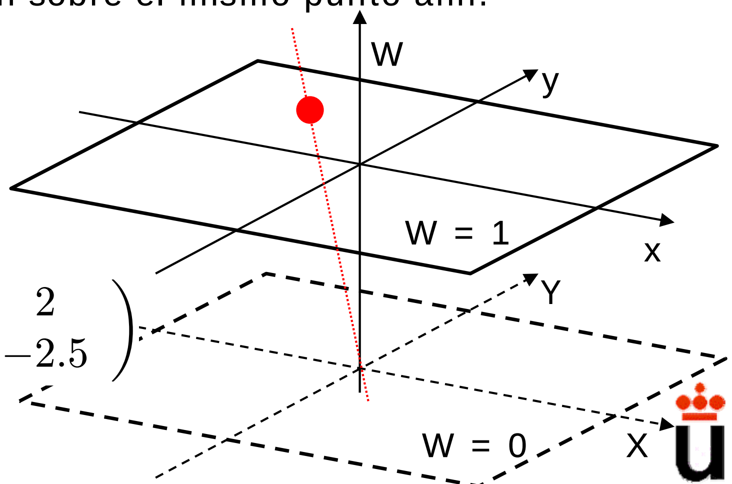


Espacios Afín y Proyectivo

- Todos los puntos de una recta que pasa por el origen se proyectan sobre el mismo punto afín.

$$\begin{pmatrix} 4 \\ -5 \\ 2 \end{pmatrix} \downarrow$$

$$\begin{pmatrix} \frac{4}{2} \\ \frac{-5}{2} \end{pmatrix} = \begin{pmatrix} 2 \\ -2.5 \end{pmatrix}$$

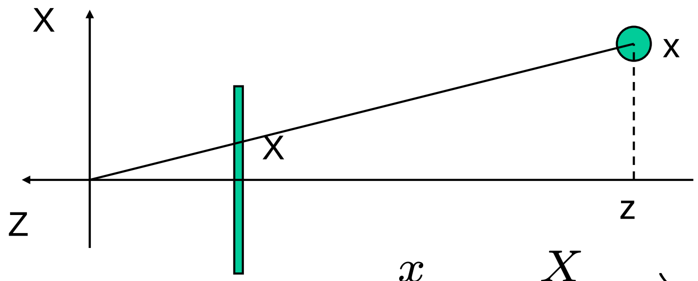


Curiosidades: Infinito

- Dos rectas paralelas del espacio afín intersectan "en el infinito". ¿Pero dos pares distintos de rectas intersectan en el mismo lugar del infinito?
- La intersección de rectas paralelas se puede analizar en el espacio proyectivo. Intersectan en un punto con w=0.
- El plano w=0 del espacio proyectivo 3D se corresponde con el infinito del espacio afín 2D.

Proyección 2D → 1D

- La posición de un punto proyectado en una imagen se puede calcular por 'triángulos semejantes'



Z = -d
plano imagen

$$\frac{x}{z} = \frac{X}{-d} \Rightarrow X = \frac{-d \cdot x}{z}$$

Proyección 2D → 1D

- Vamos a definir la proyección mediante 2 pasos:

1. Una transformación lineal del espacio afín 2D al espacio proyectivo 2D.
2. La proyección (división por W) que pasa del espacio proyectivo 2D al espacio afín 1D.

- Esta definición nos permite concatenar la transformación lineal inicial a todas las transformaciones geométricas del proceso 'modelview'.

Proyección 2D → 1D

- Para calcular la transformación, hacemos la siguiente observación: La división por z es el paso no-lineal de T, y se puede ejecutar en la división por W.

$$\begin{pmatrix} x \\ z \end{pmatrix} \xrightarrow{\text{T}} \begin{pmatrix} X' \\ W \end{pmatrix} \xrightarrow{\text{1/W}} \begin{pmatrix} X \end{pmatrix}$$

$$T = \begin{pmatrix} -d & 0 \\ 0 & 1 \end{pmatrix}$$

- Llevándolo a 3D, la coordenada y se comporta igual.

Proyección 3D → 2D

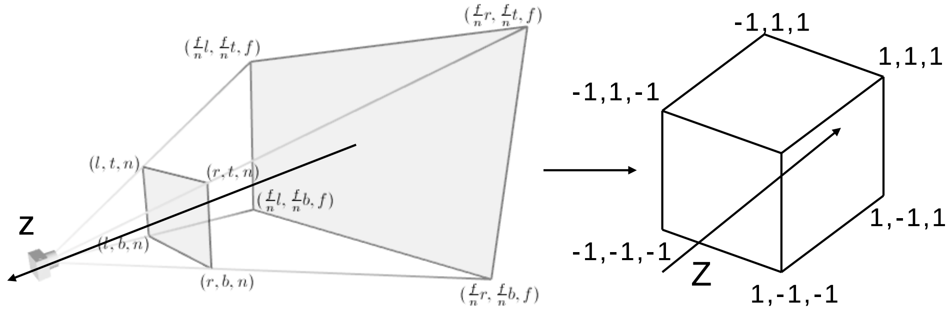
- Incluimos los términos Z y W en todo el proceso:

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \xrightarrow{\quad T \quad} \begin{pmatrix} X' \\ Y' \\ Z' \\ W \end{pmatrix} \xrightarrow{\quad 1/W \quad} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

$$T = \begin{pmatrix} -d & 0 & 0 & 0 \\ 0 & -d & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Proyección en OpenGL

- En lugar de proyectar a un plano, el *view frustum* se proyecta a un cubo.



Proyección en OpenGL

- Hay dos principales motivos:
  - Permite hacer el test de visibilidad (*z-buffer*) después de la proyección
  - Es mucho más fácil hacer el *clipping* en el cubo
- La coordenada Z se almacena en representación entera. Su resolución depende de la elección de los planos *near* y *far*, y su distribución es no-lineal, con mayor resolución cerca del plano near.

Proyección en OpenGL

- Matriz de proyección en OpenGL:

$$\mathbf{T} = \begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

- Nota: los valores de *l*, *b*, *n* y y son negativos.