



# Cámaras y visibilidad

---

Modelo de cámara virtual  
Trasformación de la vista  
Trasformación proyectiva  
Visibilidad  
Interacción con la cámara



Sección de  
Informática  
Gráfica | Computer  
Graphics  
Group  
VALENCIA



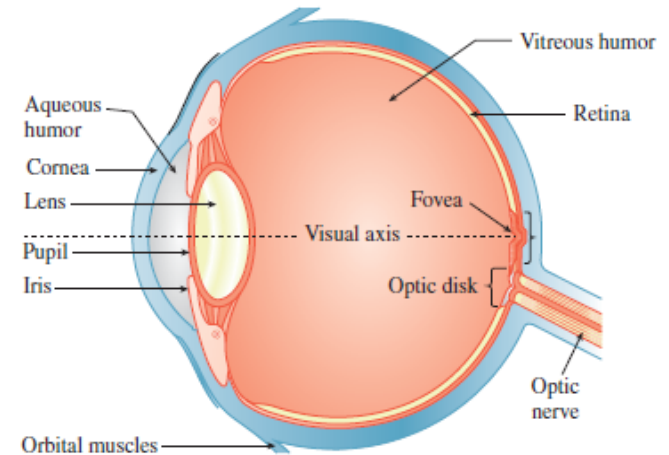
UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



# Modelo de cámara virtual

# Visión 3D

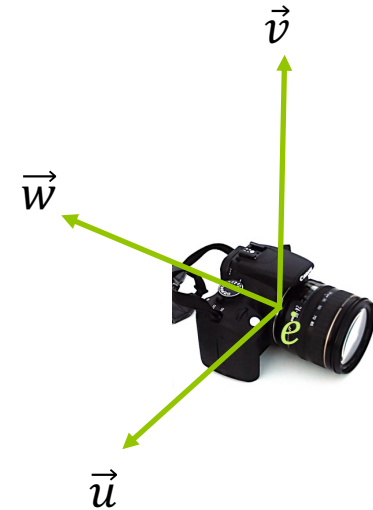
- ▶ **Visión 3D real**
  - ▶ Visión estéreo
  - ▶ Convergencia visual
  - ▶ Enfoque
  - ▶ Campo de visión
  - ▶ Perspectiva
  - ▶ Sensibilidad
- ▶ **Visión 3D simulada**
  - ▶ Intenta simular la visión real mediante el símil de una cámara fotográfica
    - ▶ Campo de visión: volumen de la vista
    - ▶ Estereoscopia: monoscopia dual
    - ▶ Perspectiva: tamaño inversamente proporcional a la distancia
    - ▶ Enfoque: Emborronado artificial de la imagen
    - ▶ Sensibilidad: Regulación de la luminosidad de la imagen
  - ▶ La cámara virtual permite simular efectos no reales



# Tipo de cámara

- ▶ Sistema de referencia de la cámara o vista
- ▶ Tipo de la cámara
  - ▶ Viene dado por el volumen de la vista (frustum)
  - ▶ El frustum se delimita por 6 planos referidos al sistema de referencia de la cámara
    - ▶ Cámara ortográfica: Ortoedro
    - ▶ Cámara perspectiva: Pirámide truncada

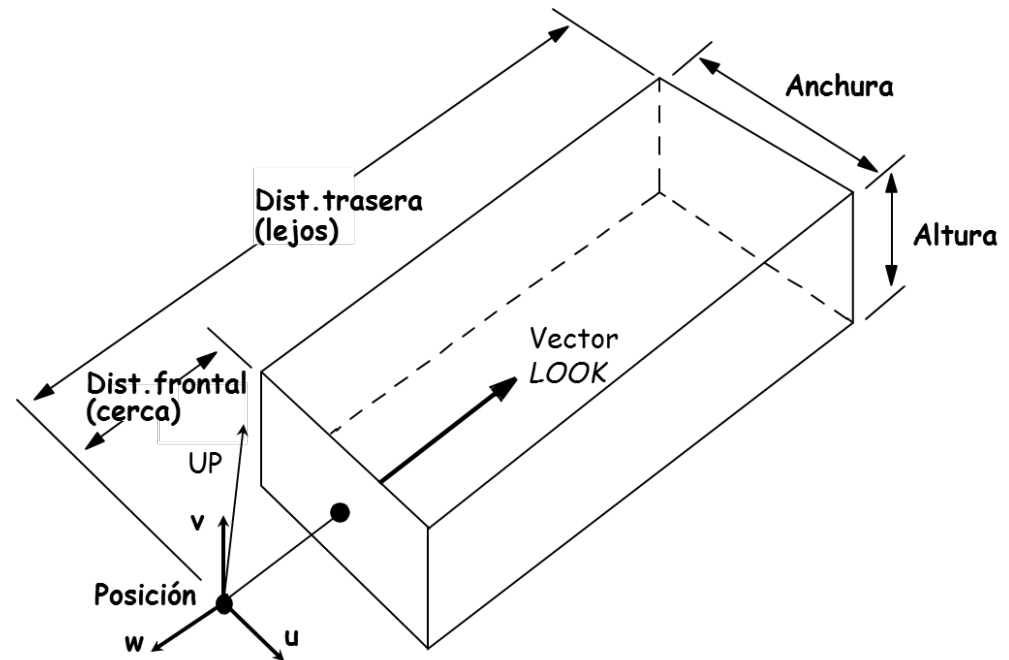
$$\vec{e}^T = [\vec{u} \quad \vec{v} \quad \vec{w} \quad e]$$



# Cámara ortográfica

## ▶ Parámetros del frustum

- ▶ ancho
- ▶ alto
- ▶ cerca
- ▶ lejos



*planos  
delimitadores*

$$\vec{u} \quad \begin{aligned} l &= -\text{ancho}/2 \\ r &= \text{ancho}/2 \end{aligned}$$

$$\vec{v} \quad \begin{aligned} b &= -\text{alto}/2 \\ t &= \text{alto}/2 \end{aligned}$$

$$\vec{w} \quad \begin{aligned} n &= -\text{cerca} \\ f &= -\text{lejos} \end{aligned}$$

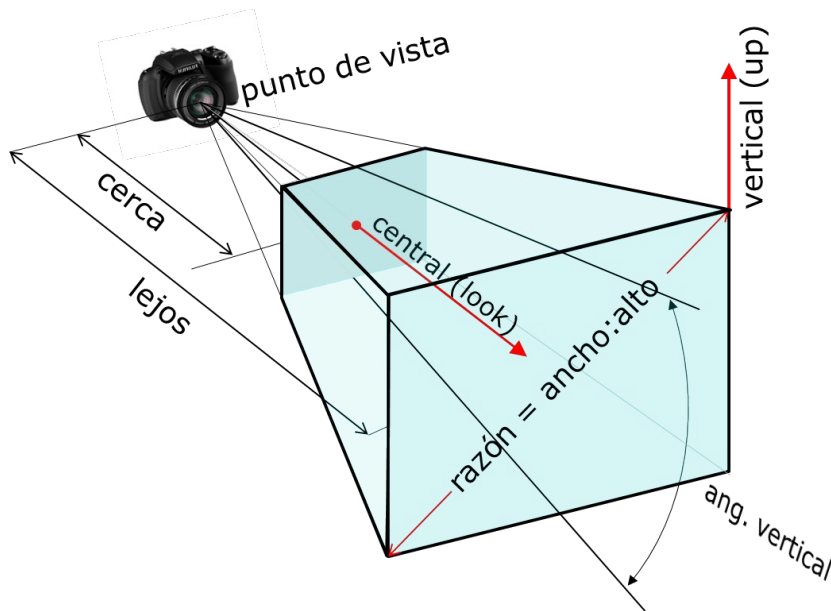
constructor

`CamaraOrtografica(l,r,b,t,n,f);`

# Cámara perspectiva

## ▶ Parámetros del frustum

- ▶ ángulo vertical de apertura
- ▶ razón de aspecto
- ▶ cerca, lejos



razón de aspecto

$$\frac{w}{h} = \frac{tg(\frac{1}{2} \Theta_w)}{tg(\frac{1}{2} \Theta_h)}$$

constructor

`CamaraPerspectiva(fovy, ar, cerca, lejos);`

*rectángulo en  
el plano cercano*

$$t = cerca \cdot \tan(fovy/2)$$

$$b = -t$$

$$r = ar \cdot t$$

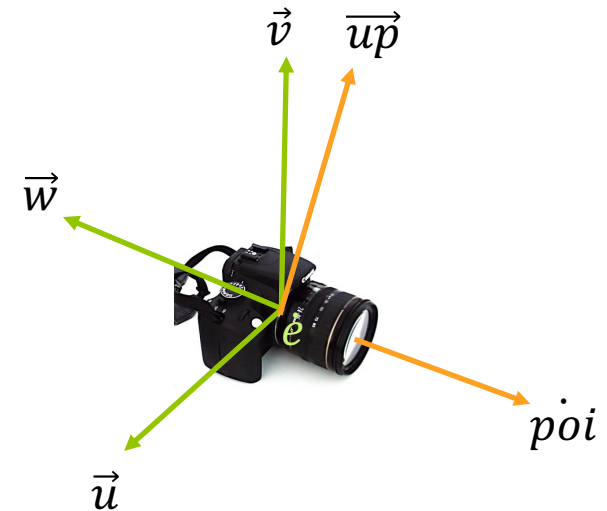
$$l = -r$$

$$n = -cerca$$

$$f = -lejos$$

# Posición y orientación de la cámara

- ▶ Parámetros para situar la cámara
  - ▶  $e$ : posición del ojo
  - ▶  $poi$ : punto de interés, hacia dónde se mira
  - ▶  $up$ : vertical subjetiva
- ▶ Las coordenadas de los parámetros se indican en el sistema de referencia del mundo
- ▶ Sistema de referencia de la vista



$$\vec{e}^T = [\vec{u} \quad \vec{v} \quad \vec{w} \quad e]$$

$$\vec{w} = (\vec{e} - poi).normalized$$

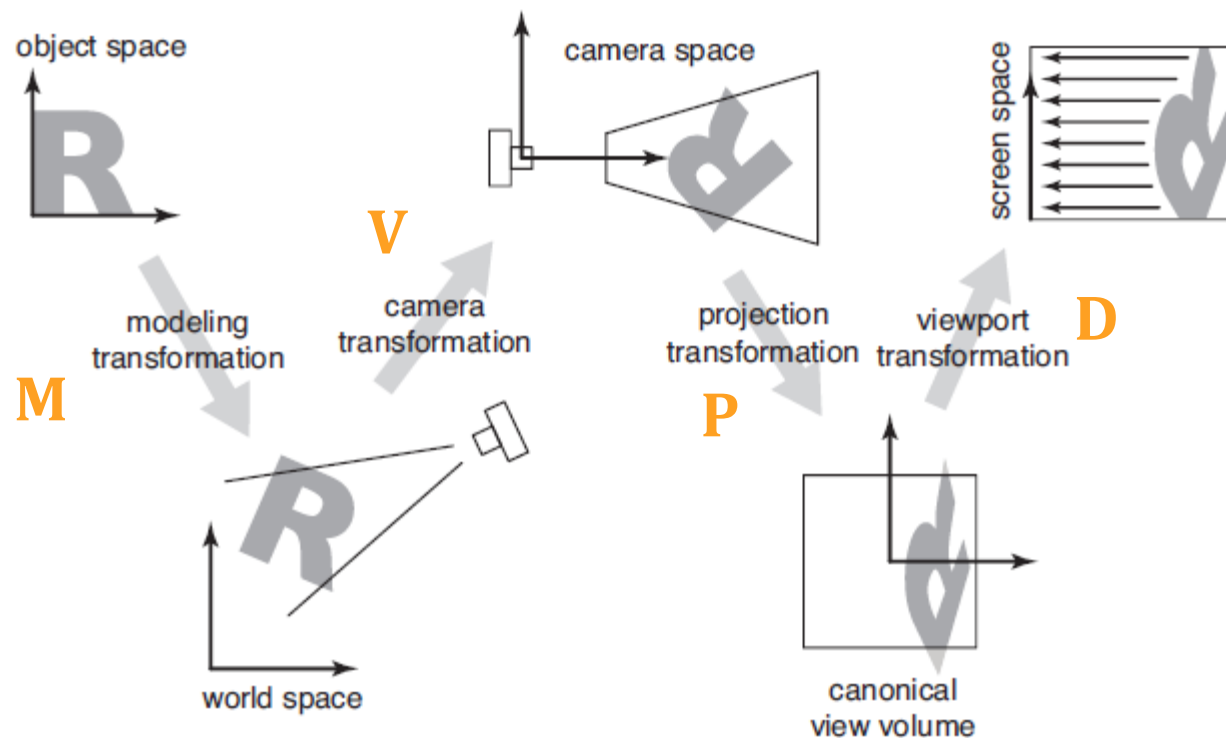
$$\vec{u} = (\vec{up} \times \vec{w}).normalized$$

$$\vec{v} = \vec{w} \times \vec{u}$$

método

```
camara.lookAt( ex,ey,ez
                poix,poiy,poiz,
                upx,upy,upz );
```

# Cadena de transformaciones M-V-P-D





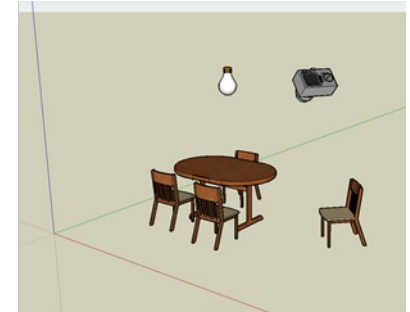


# Trasformación de la vista

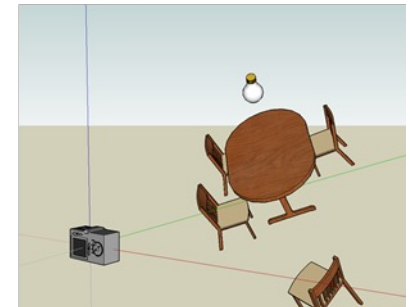
# Trasformación de la vista (V)

- ▶ Dadas las coordenadas de un punto en el sistema del mundo real queremos conocer sus coordenadas en el sistema de la cámara
- ▶ La matriz  $V$  es la composición de un traslación y un giro

$\vec{w}^T$



$\vec{e}^T$



$$V = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -e_x \\ 0 & 1 & 0 & -e_y \\ 0 & 0 & 1 & -e_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

giro para bases  
coincidentes

traslación para  
orígenes coincidentes

$$\dot{p} = \vec{w}^T p = \vec{e}^T V p$$

# Inversa de la transformación de la vista

- ▶ Coordenadas del mundo conocidas las de la vista
- ▶ Operación inversa: giro inverso y traslación inversa

$$\dot{p} = \vec{w}^T V^{-1} p_e = \vec{e}^T p_e$$

inversa = traspuesta

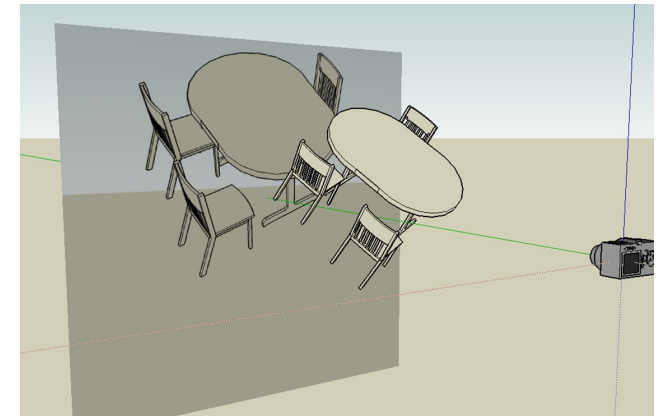
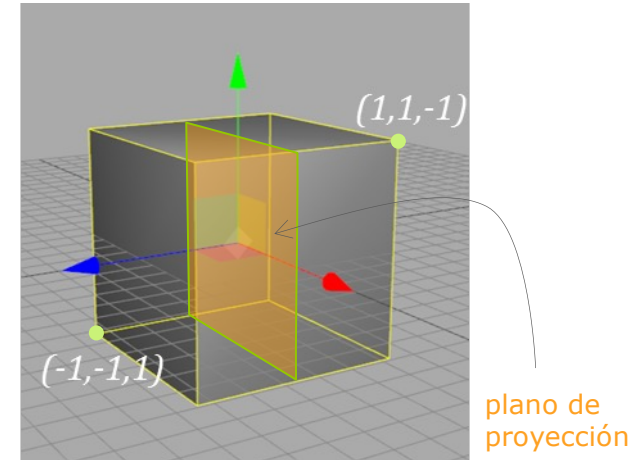
$$V^{-1} = \begin{bmatrix} 1 & 0 & 0 & e_x \\ 0 & 1 & 0 & e_y \\ 0 & 0 & 1 & e_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 & e_x \\ 0 & 1 & 0 & e_y \\ 0 & 0 & 1 & e_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} u_x & v_x & w_x & e_x \\ u_y & v_y & w_y & e_y \\ u_z & v_z & w_z & e_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



# Trasformación proyectiva

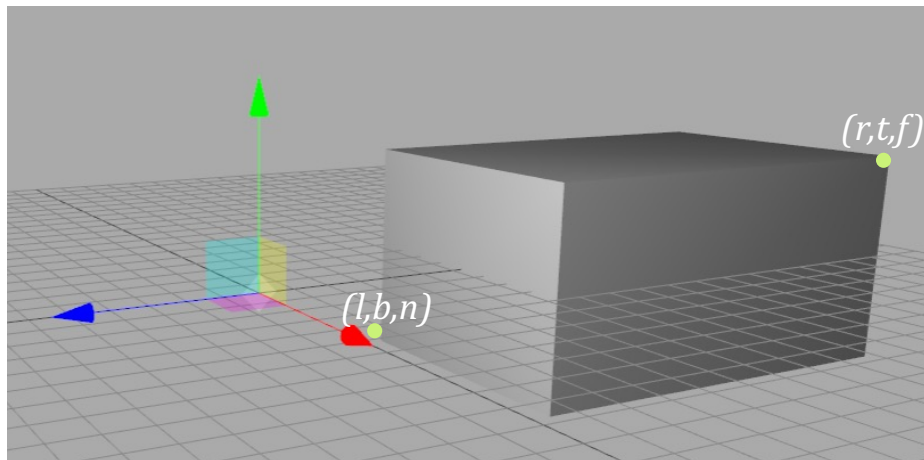
# Trasformación proyectiva

- ▶ Sistema canónico de la proyección
- ▶ Transformación proyectiva
- ▶ Proyección:  $(x,y,z)$
- ▶ Vista proyectiva a ortográfica

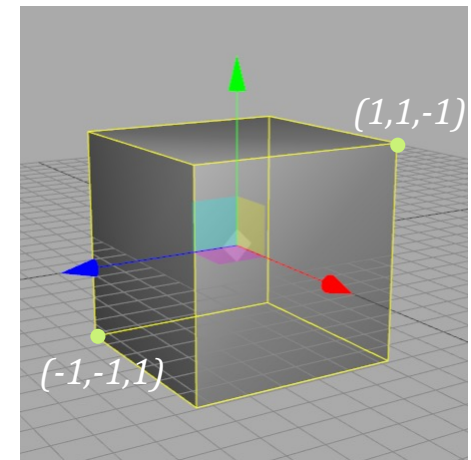


# Trasformación proyectiva ortográfica

- ▶ Frustum canónico: cubo de lado 2 centrado en el origen
- ▶ Trasformación del frustum ortográfico al canónico



frustum ortográfico



frustum canónico

# Matriz de proyección ortográfica

1. Traslación para que coincidan los centros de los ortoedros
2. Escalado al cubo canónico

Diagram illustrating the construction of the orthographic projection matrix  $P_{orto}$  through two steps:

- Step 1 (Translation):** Translating the object so its center coincides with the origin. The translation vector is  $(r-l)/2, (t+b)/2, (n+f)/2$ . The scaling factor is  $2, 2, 2$ .
- Step 2 (Scaling):** Scaling the object to fit the canonical cube. The scaling factors are  $0, 0, 0$ .

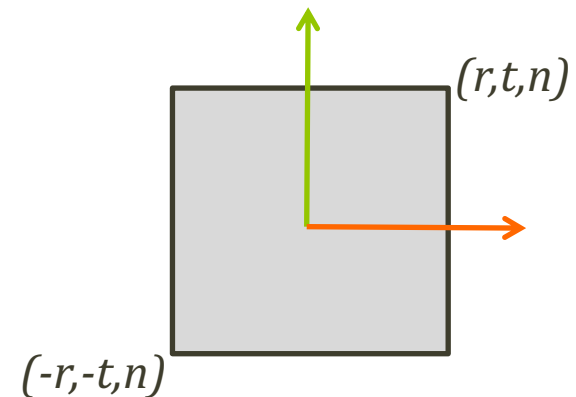
$$P_{orto} = \begin{bmatrix} 2/r-l & 0 & 0 & 0 \\ 0 & 2/t-b & 0 & 0 \\ 0 & 0 & 2/n-f & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -(r+l)/2 \\ 0 & 1 & 0 & -(t+b)/2 \\ 0 & 0 & 1 & -(n+f)/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2/r-l & 0 & 0 & -(r+l)/(r-l) \\ 0 & 2/t-b & 0 & -(t+b)/(t-b) \\ 0 & 0 & 2/n-f & -(n+f)/(n-f) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Las coordenadas del punto proyectado  
son las coordenadas  $(x,y)$  después de aplicar  $P$

# Matriz de proyección ortográfica

- ▶ Cuando el volumen de la vista está centrado ( $r=-l$ ;  $t=-b$ ) la matriz de proyección ortográfica se reduce a:

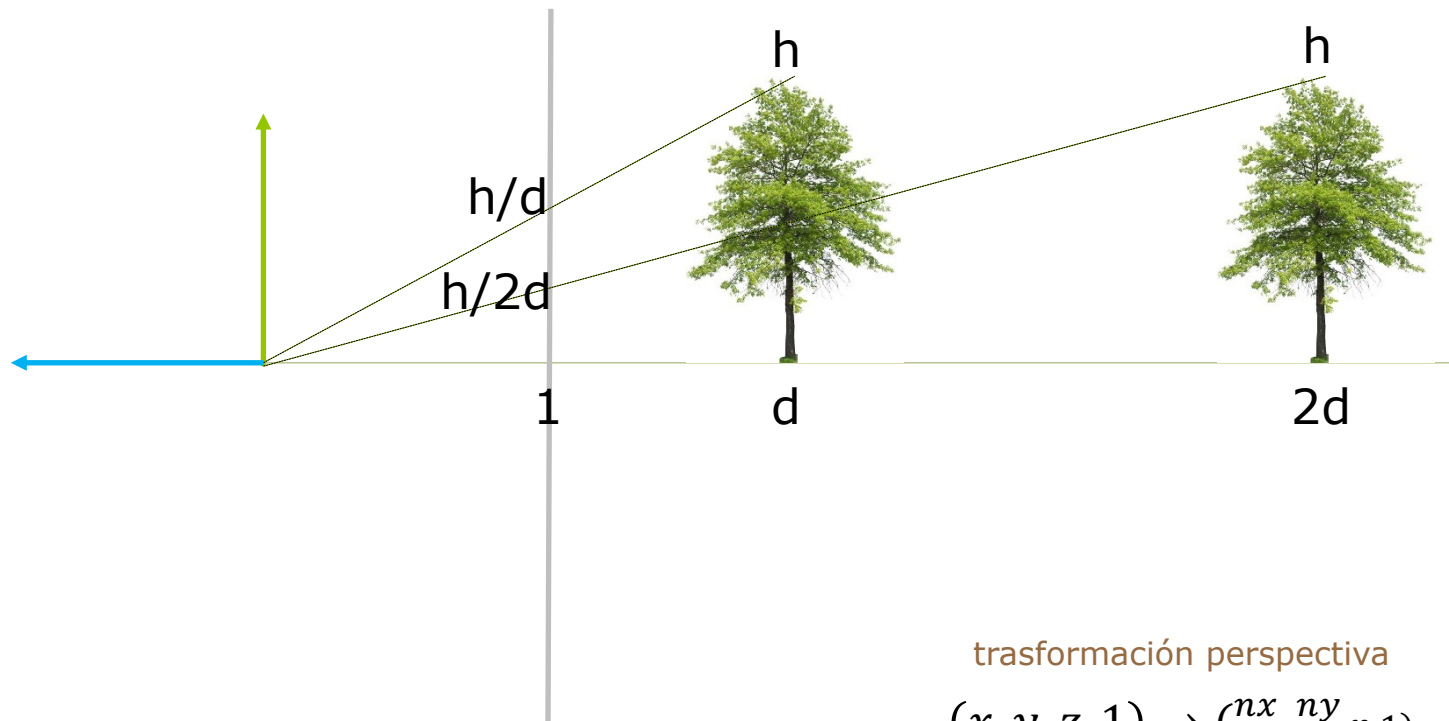
$$P_{orto} = \begin{bmatrix} 1/r & 0 & 0 & 0 \\ 0 & 1/t & 0 & 0 \\ 0 & 0 & 2/(n-f) & -(n+f)/(n-f) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$





# Proyección perspectiva

- ▶ En la proyección perspectiva el tamaño es inversamente proporcional a la distancia al observador

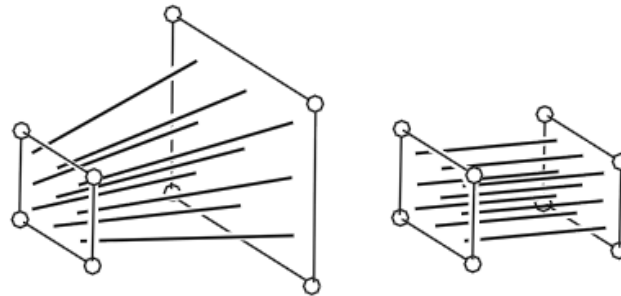


transformación perspectiva

$$(x, y, z, 1) \rightarrow \left(\frac{nx}{z}, \frac{ny}{z}, n, 1\right)$$

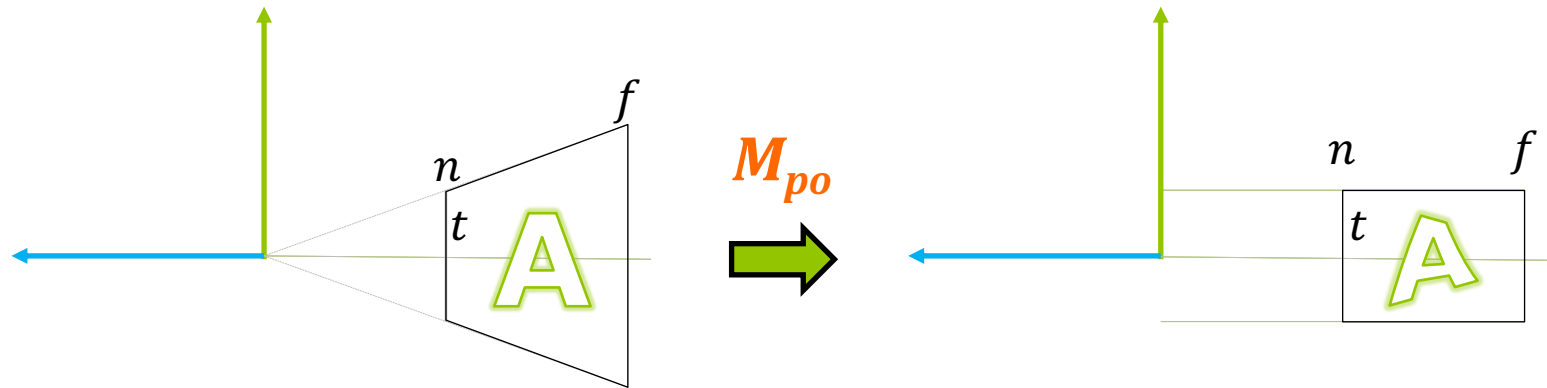
# Trasformación proyectiva perspectiva

1. Convertimos el volumen de la vista perspectiva en un ortoedro deformando los objetos



- A. La ordenación en distancia se conserva
  - B. Lo que estaba dentro sigue dentro y al revés
  - C. La proyección ortográfica es la misma que la proyección en perspectiva
2. Aplicamos la división perspectiva
    1. La transformación entre ambos volúmenes no se puede realizar aplicando una única matriz
    2. Se necesita normalizar la cuarta coordenada

# Matriz perspectiva-ortográfica



## condiciones

$x:$

$$x = \frac{r}{n}z \Rightarrow x' = r$$

$y:$

$$y = \frac{t}{n}z \Rightarrow y' = t$$

$z:$

$$z \rightarrow 0^- \Rightarrow z' \rightarrow \infty$$

$$z = n \Rightarrow z' = n$$

$$z = f \Rightarrow z' = f$$

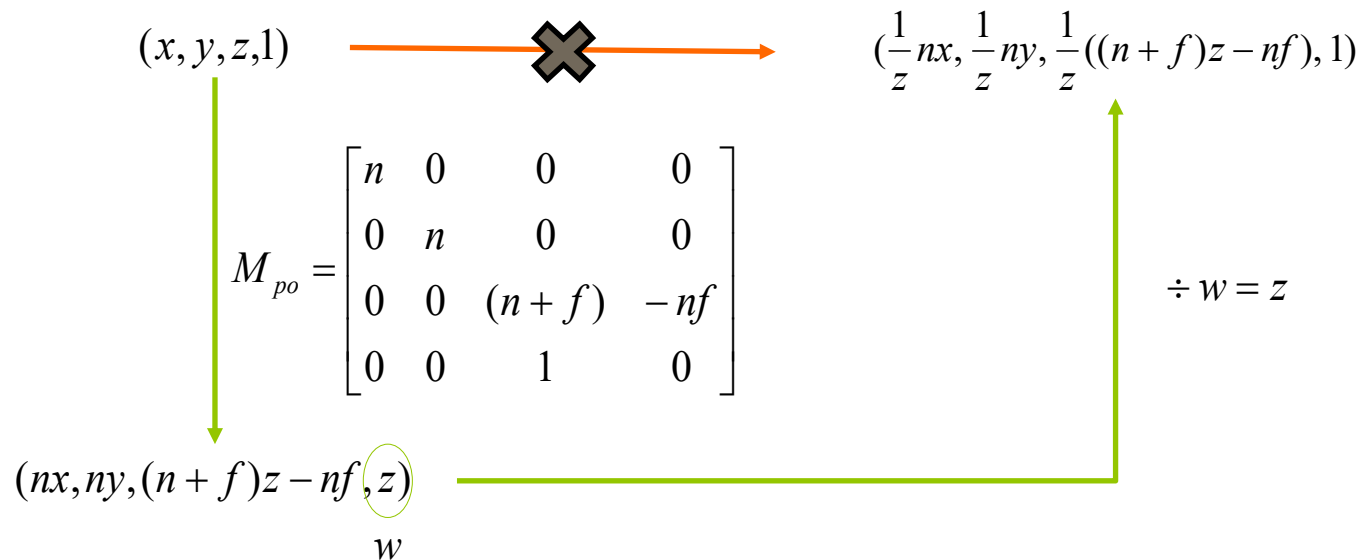
## transformación

$$(x, y, z, 1) \rightarrow$$

$$\left(\frac{1}{z}nx, \frac{1}{z}ny, \frac{1}{z}((n+f)z - nf), 1\right)$$

# Matriz perspectiva-ortográfica

- ▶ No podemos usar una matriz directa para la transformación
- ▶ Aplicaremos una matriz proyectiva y después la división perspectiva



# Trasformación al cubo canónico

- ▶ Una vez en ortográfica transformaremos al cubo canónico
- ▶ Usaremos la matriz de proyección ortográfica con el volumen de la vista centrado
- ▶ La división perspectiva puede hacerse después de esta transformación

$$P_{orto} M_{po} = \begin{bmatrix} n/r & 0 & 0 & 0 \\ 0 & n/t & 0 & 0 \\ 0 & 0 & n+f/n-f & -2fn/n-f \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

coordenadas del punto  
en el sistema del cubo  
canónico

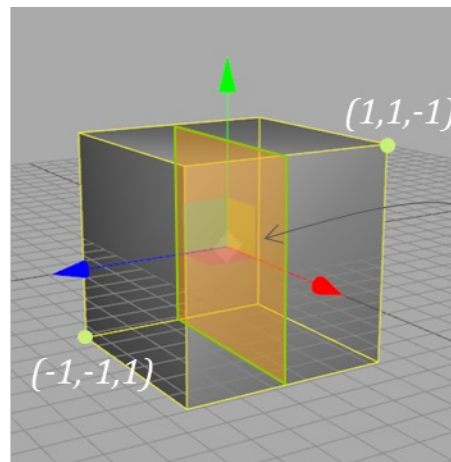
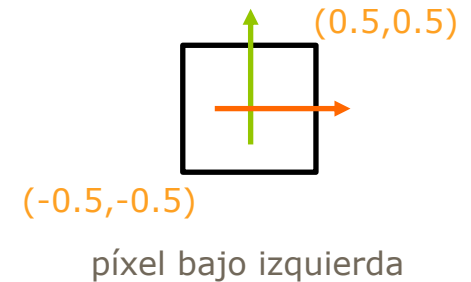
coordenadas del punto  
en el sistema de cámara  
ortográfica

coordenadas del punto  
antes de deformar  
en el sistema de cámara  
perspectiva

$$\vec{c}^T \mathbf{r} = \vec{c}^T P_{orto} \mathbf{p} = \vec{c}^T P_{orto} \frac{1}{p''_w} \mathbf{M}_{po} \mathbf{q} = \vec{c}^T \frac{1}{p''_w} P_{orto} \mathbf{M}_{po} \mathbf{q}$$

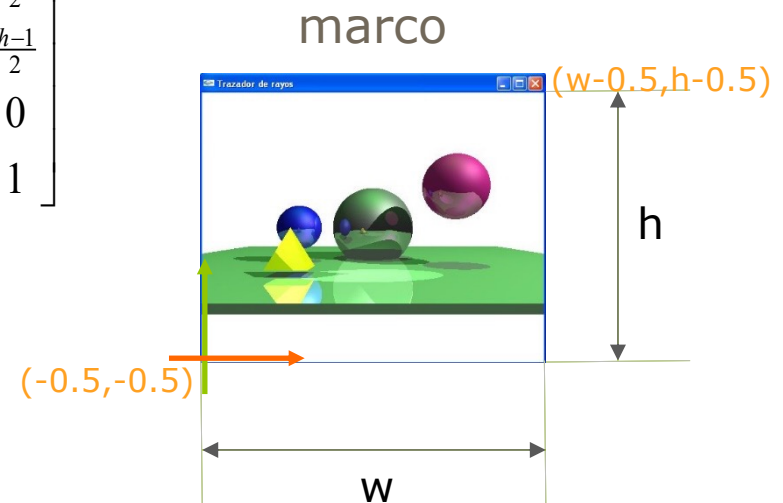
# Trasformación del marco

- ▶ Se hace corresponder la ventana de proyección con el marco de representación en el dispositivo
- ▶ El marco es un rectángulo de  $w \times h$  píxeles. Las coordenadas enteras caen en el centro de los píxeles
- ▶ Se recupera la isotropía si las relaciones de aspecto son equivalentes
- ▶ Es la composición de un escalado y una traslación (2D)

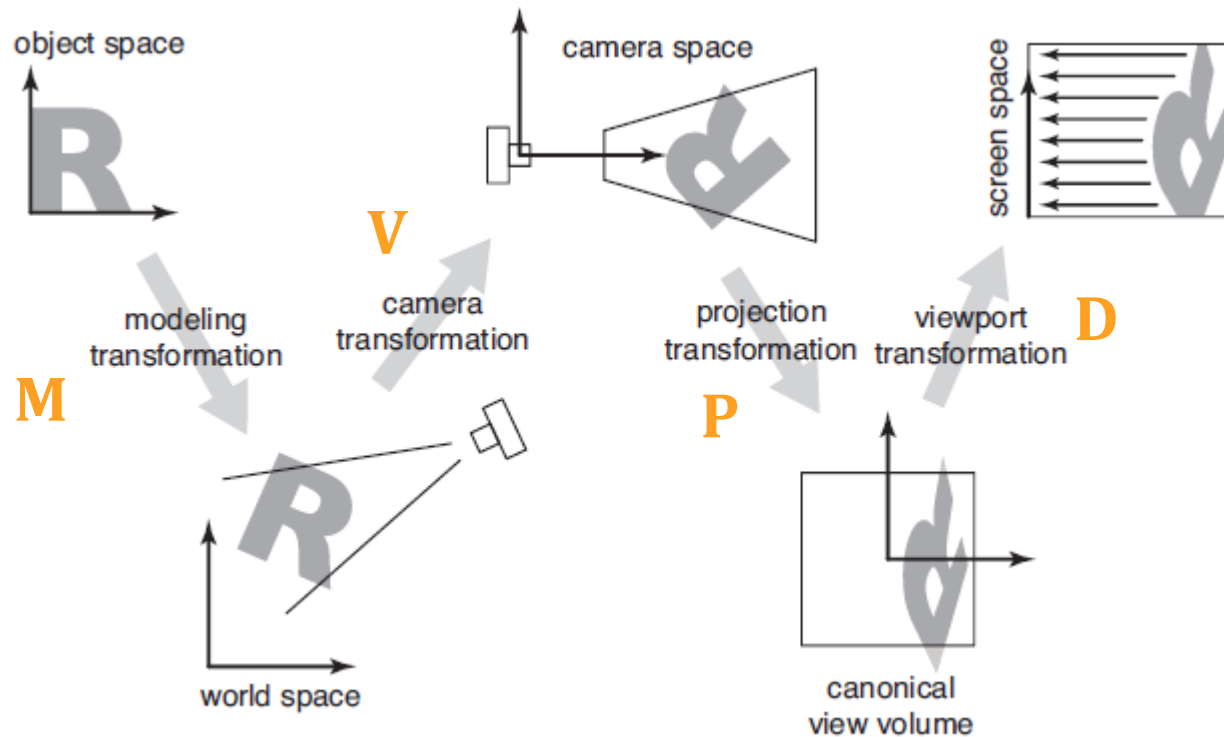


plano de  
proyección  
2x2

$$D = \begin{bmatrix} w/2 & 0 & 0 & \frac{w-1}{2} \\ 0 & h/2 & 0 & \frac{h-1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



# Cadena de transformaciones M-V-P-D



$$\vec{d}^T p' = \vec{d}^T \frac{1}{p''_w} p'' = \vec{d}^T \frac{1}{p''_w} D P_{orto} M_{po} V M p$$



Sección de  
Informática  
Gráfica | Computer  
Graphics  
Group  
VALENCIA



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



# Visibilidad



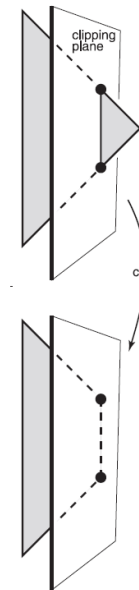
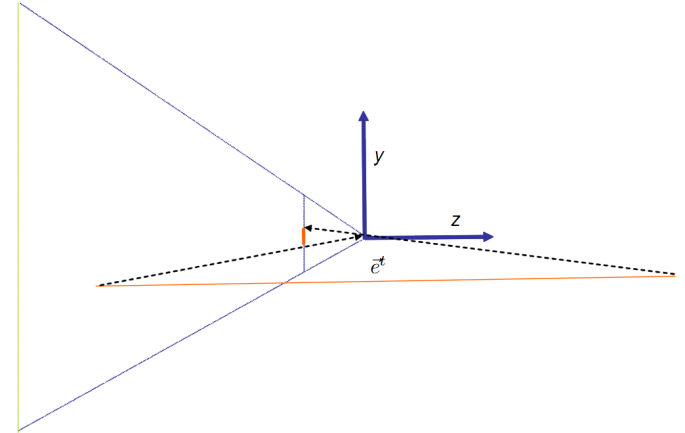
# Visibilidad

---

- ▶ ¿Qué objetos son visibles una vez elegida y situada la cámara?
- ▶ Por eficiencia, sería deseable solo muestrear aquellas partes visibles (etapa de rasterización)
- ▶ No son visibles si:
  1. No están dentro del volumen de la vista: proceso de recortado (**clipping**)
  2. Es un polígono de una superficie cerrada que da la espalda al observador: eliminación de caras traseras (**culling**)
  3. Es un polígono diferente al caso anterior pero está tapado por otro: visibilidad por cercanía (**visibility**)

# ¿Dentro del frustum?

- ▶ El proceso de recortado consiste en eliminar las partes o el total de un polígono respecto a un volumen
- ▶ Nuestro volumen es el de la vista
  - ▶ Es convexo
  - ▶ Después de la transformación perspectiva es el cubo canónico
- ▶ El recortado se suele realizar antes de la división perspectiva, en el espacio 4D (espacio de recortado)
- ▶ Los atributos por vértice se dividen por la z y se interpolan linealmente:  $a' = -a/z$
- ▶ Los polígonos resultantes se teselan a triángulos



*Condiciones de interioridad de un punto al hipercubo*

$$w > x > -w \ \&$$

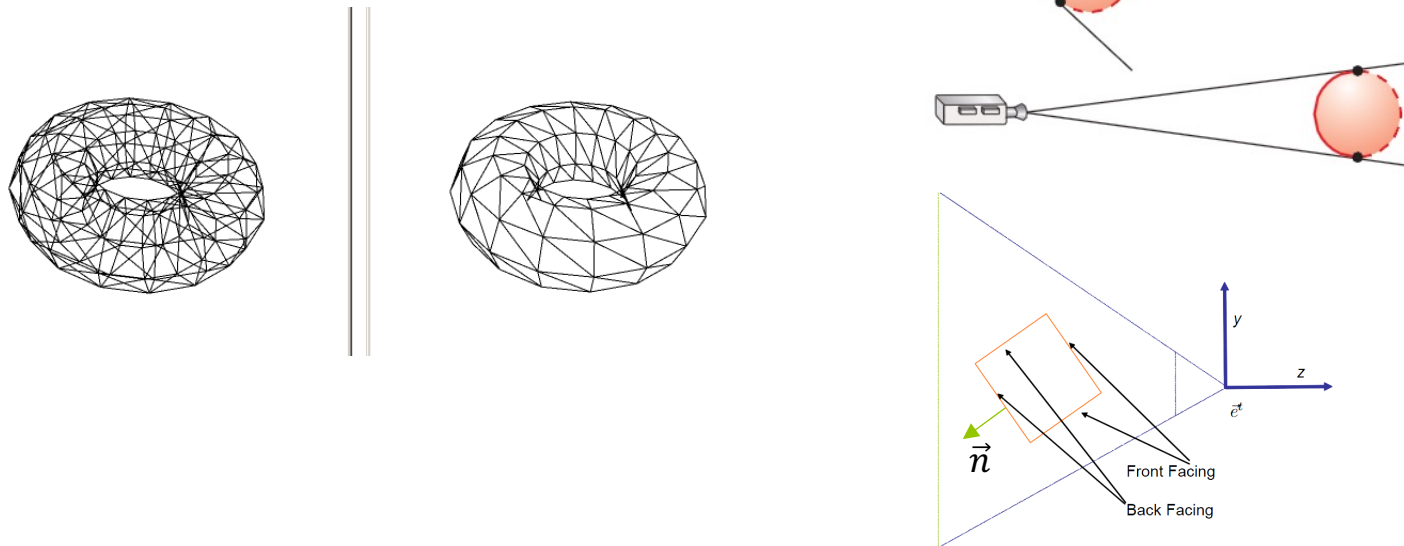
$$w > y > -w \ \&$$

$$w > z > -w$$

Ojo: si  $w < 0$  es al revés

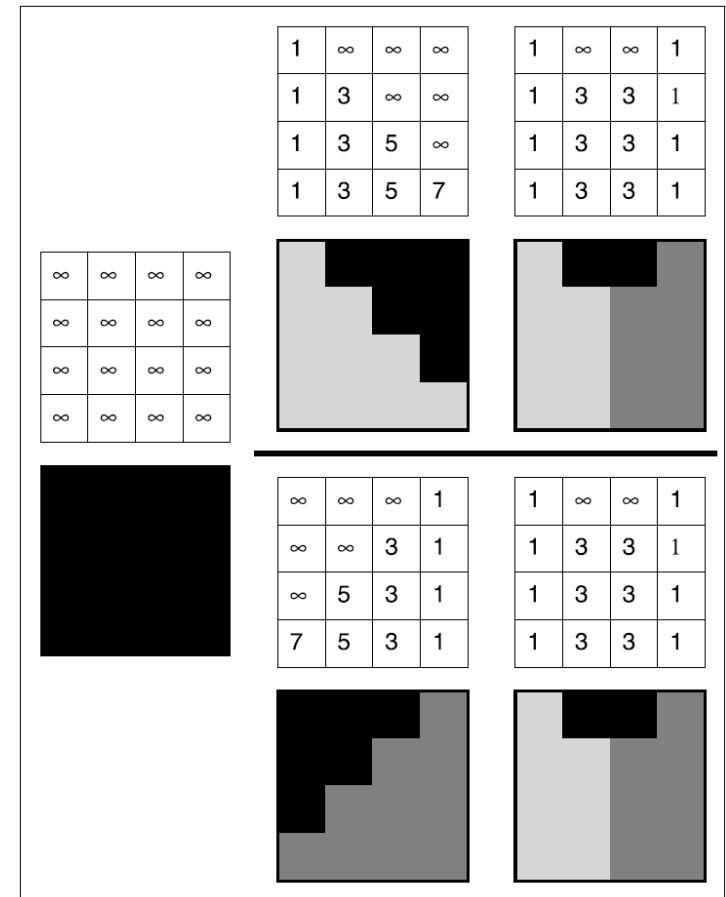
# ¿Está de espaldas?

- ▶ Eliminación de caras traseras
  - ▶ Simplemente se comprueba la normal en el sistema de referencia de la cámara. Si  $n_z < 0$  es trasera.
  - ▶ Sólo sirve para cuerpos sólidos y opacos
  - ▶ Conveniente cuando se usa representación alámbrica



# Delante y detrás

- ▶ Dos soluciones
  - ▶ Pre-ordenamiento (pintor)
  - ▶ Sin importar el orden (z-buffer)
- ▶ Z-Buffer
  - ▶ En combinación con el proceso de rasterización se mantiene un buffer de profundidad para los fragmentos procesados descartando aquellos que están más alejados que el actual
  - ▶ Aliasing espacial
  - ▶ Aliasing numérico (Z-fighting)





Sección de  
Informática  
Gráfica | Computer  
Graphics  
Group  
VALENCIA



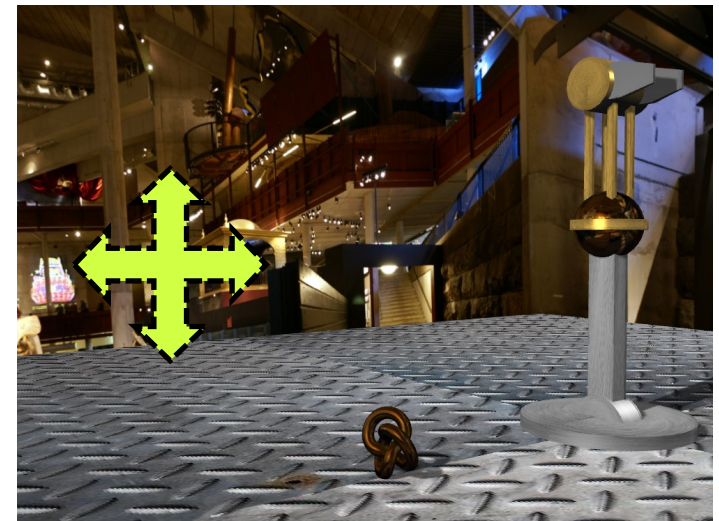
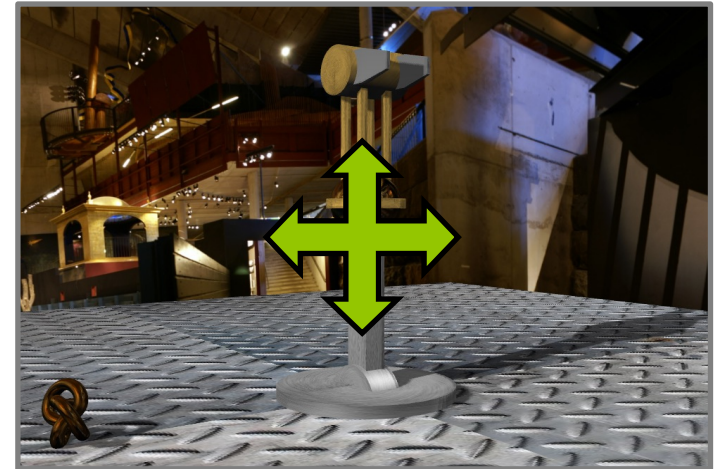
UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



# Interacción con la cámara

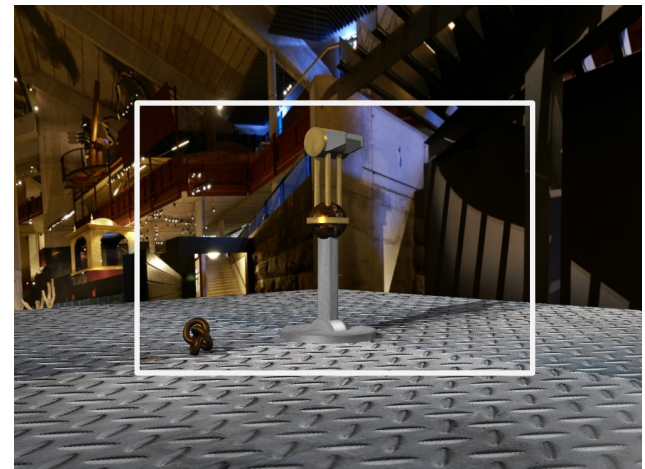
# Desplazamiento (panning)

- ▶ Movimiento de la cámara paralelamente al plano de proyección manteniendo la dirección central de la vista
- ▶ Se mantiene
  - ▶ La base del sistema de referencia de la vista  $\vec{u}, \vec{v}, \vec{w}$
  - ▶ Vectores  $\overrightarrow{up}$  y  $\overrightarrow{look}$
  - ▶ El volumen de la vista
- ▶ Cambia
  - ▶ El origen del sistema de referencia de la vista (*eye*) mediante una traslación en  $\vec{u}, \vec{v}$
  - ▶ El punto de interés (*poi*)



# Acercar/Alejar (zoom)

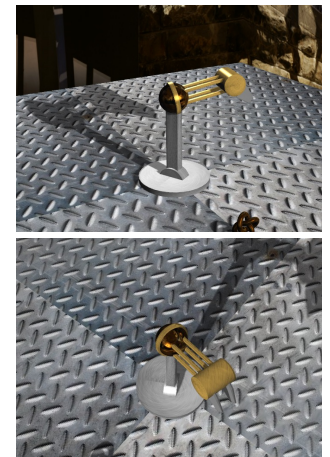
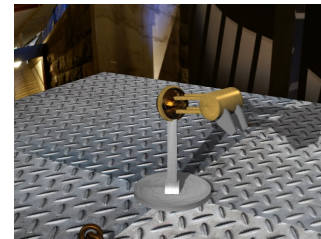
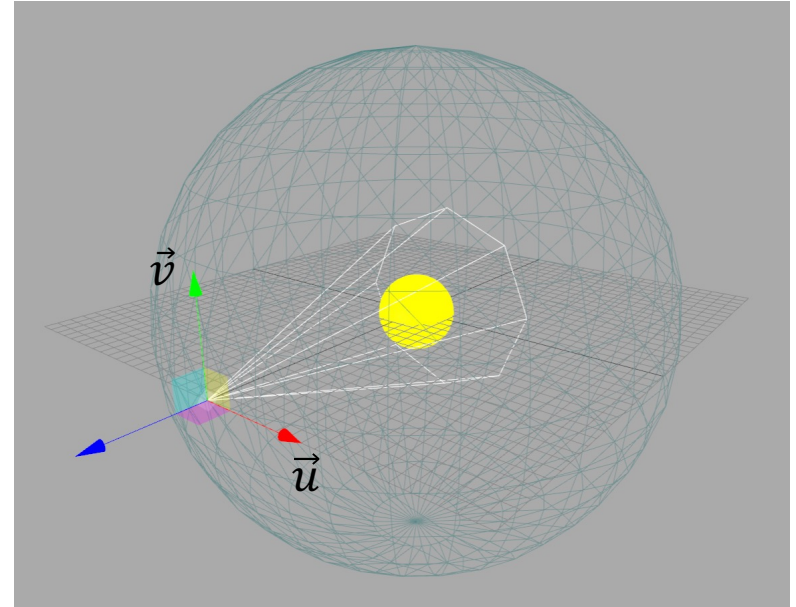
- ▶ Consiste en aumentar o disminuir el tamaño de los objetos en la imagen final
- ▶ Se consigue
  1. Variando la distancia entre la cámara y el punto de interés
    - ▶ Se varía la posición del ojo (*eje*) en el eje  $\vec{w}$
    - ▶ Todo lo demás se mantiene
  2. Variando campo de vista vertical
    1. Varía el *fovy* sin modificar la distancia entre la cámara y el motivo
    2. Todo lo demás se mantiene





# Orbital

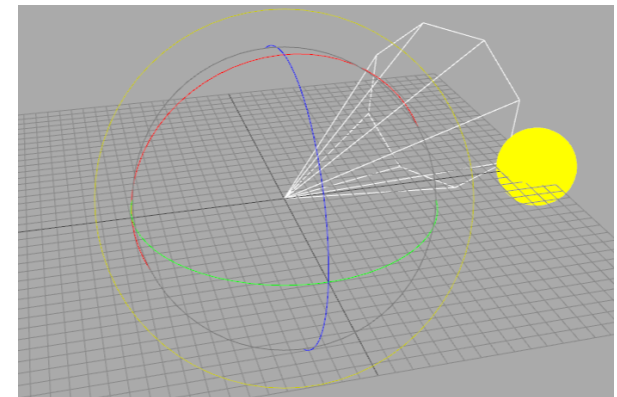
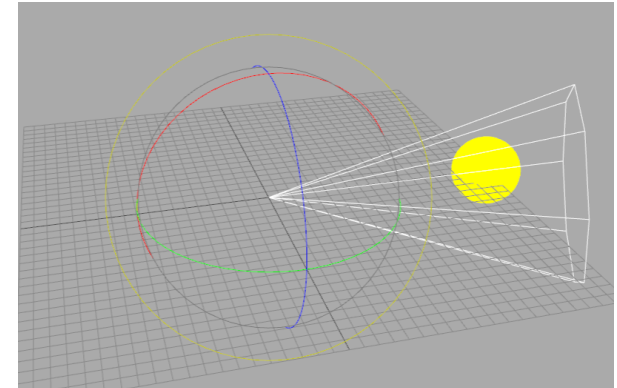
- ▶ La cámara se mueve sobre una esfera imaginaria que envuelve al motivo
- ▶ Varía
  - ▶ La posición y orientación de la cámara
- ▶ Se mantiene
  - ▶ El punto de interés, el frustum y la distancia entre el ojo y el motivo
- ▶ Tipos de movimientos
  - ▶ Arriba/abajo
    - ▶ Giro alrededor del eje  $\vec{u}$  en el sistema de referencia del mundo
  - ▶ Izquierda/derecha
    - ▶ Giro alrededor del eje  $\vec{v}$  en el sistema de referencia del mundo
- ▶ A veces se intercambian los sentidos de giro (inspección)





# Giro de cabeza

- ▶ El observador inspecciona su entorno sin moverse del sitio
- ▶ Se mantiene
  - ▶ La posición de la cámara y el frustum
- ▶ Varía
  - ▶ La orientación de la cámara (ejes  $\vec{u}$ ,  $\vec{v}$ ,  $\vec{w}$ )
- ▶ Tipos de movimientos
  - ▶ Bajar/levantar la cabeza
    - ▶ Giro alrededor de  $\vec{u}$  en el sistema de referencia de la cámara
  - ▶ Mirar a derecha/izquierda
    - ▶ Giro alrededor de  $\vec{v}$  en el sistema de referencia de la cámara
  - ▶ Inclinar la cabeza hacia los hombros
    - ▶ Giro alrededor de  $\vec{w}$  en el sistema de referencia de la cámara



# Desenfoque

- ▶ Consiste en fijar una distancia al motivo que se desea ver nítido y desenfocar (hacer borroso) los objetos conforme de alejan del motivo principal
- ▶ En gráficos por computador se mueve ligeramente la cámara manteniendo el plano enfocado como sección del frustum

