



Tubería Gráfica: El proceso de visualización

Rendering

Cámara sintética

Procesado de polígonos

Procesado de rayos



Sección de
Informática
Gráfica | Computer
Graphics
Group
VALENCIA



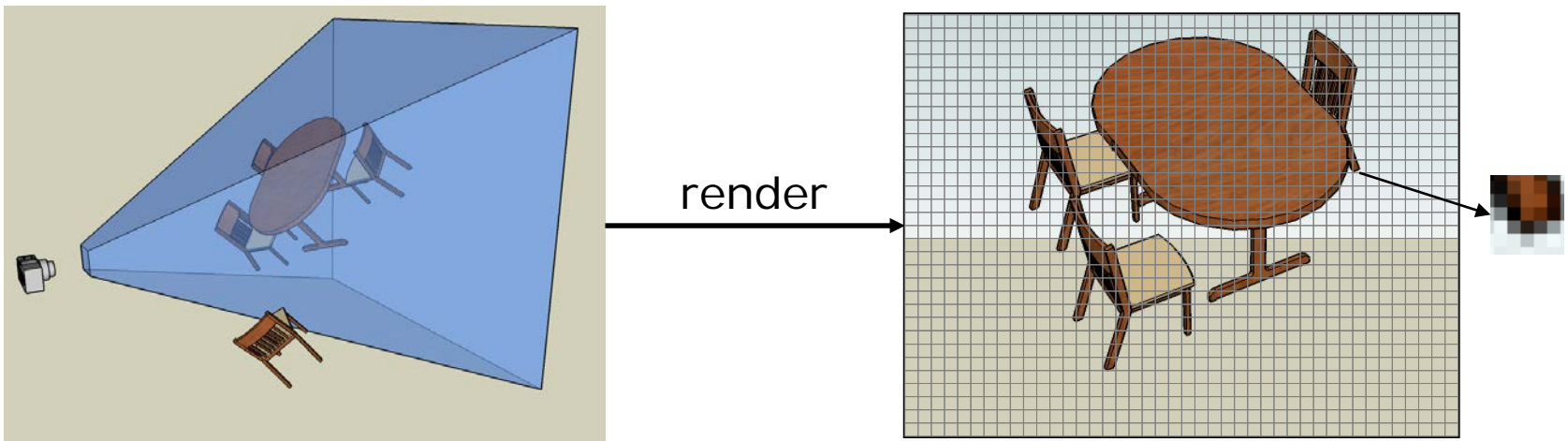
UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Rendering

Concepto de rendering

- ▶ Síntesis de una imagen a partir de información geométrica y física
- ▶ Problema a resolver: ¿Cuánta luz llega a cada sensor de la cámara?
 - ▶ No se pueden calcular las trayectorias y energías de todos los fotones
 - ▶ La solución es muestrear y reconstruir



Elementos de una escena

▶ Estática

- ▶ Objetos gráficos
 - ▶ Información geométrica
 - ▶ Información física
- ▶ Observador
 - ▶ Información geométrica
- ▶ Fuentes de luz
 - ▶ Información geométrica
 - ▶ Información física

▶ Dinámica

- ▶ Además, el tiempo
 - ▶ Influencia del tiempo en la escena: Fase de actualización (update)
 - ▶ Control del tiempo: Coherencia temporal, FPS

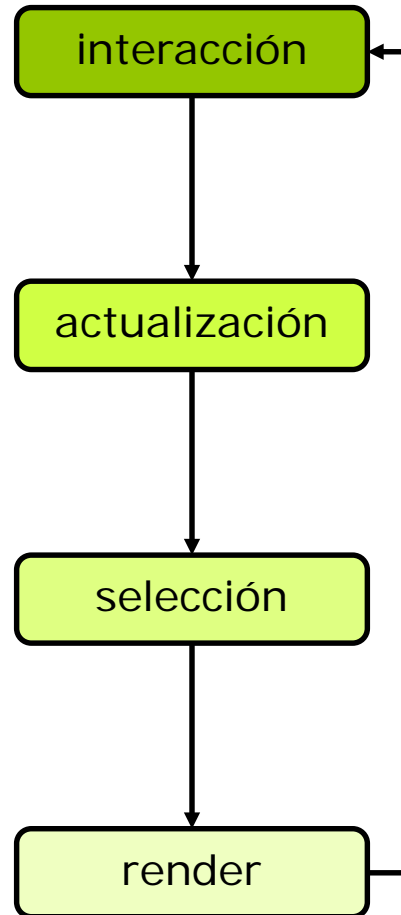
Bucle de animación

Se capturan los
eventos de entrada
(usuario, tiempo,
sensores, etc)

Se modifica la escena
en consecuencia

Se descartan aquellos
objetos que no influyen
en el render

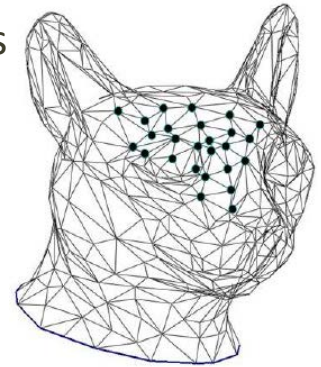
Se convierte la
información en imagen



Protagonistas del render

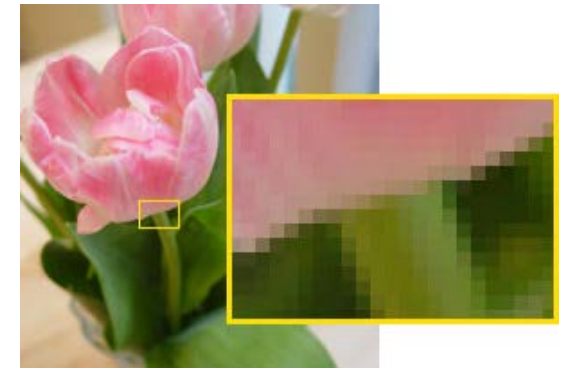
▶ Vértices

- ▶ Punto en el espacio 3D sobre la superficie de los objetos
- ▶ Es el elemento básico para construir polígonos enlazando por aristas
- ▶ Atributos del vértice
 - ▶ Coordenadas de posición (generalmente en el sistema del modelo)
 - ▶ Color (definido o calculado)
 - ▶ Vector normal (a la superficie en ese punto)
 - ▶ Coordenadas de textura (sobre el sistema de la textura)
 - ▶ Otros: tangentes, temperatura, etc



▶ Fragmentos

- ▶ Cada una de las áreas rectangulares de las que se compone la imagen. Píxeles eventuales
- ▶ Atributos:
 - ▶ Coordenadas de posición (en el espacio de la imagen)
 - ▶ Profundidad (distancia a la cámara)
 - ▶ Color
 - ▶ Coordenada de textura (en el espacio de la textura)
 - ▶ Otros: normal, tangente, temperatura, etc



Estrategias de render

- ▶ **Primero los vértices (rasterizing)**
 - ▶ Se procesan los polígonos en cualquier orden (vértice a vértice)
 - ▶ Se calculan los fragmentos a partir de la proyección de los vértices y sus atributos
 - ▶ Se procesan los fragmentos
 - ▶ Se muestra la imagen
- ▶ **Primero los fragmentos (trazado de rayos)**
 - ▶ Se calcula la visual que pasa por cada fragmento
 - ▶ Se calcula la intersección de la visual con el polígono más cercano
 - ▶ Usando los atributos de los vértices se calculan los atributos del fragmento
 - ▶ Se muestra la imagen



Sección de
Informática
Gráfica | Computer
Graphics
Group
VALENCIA



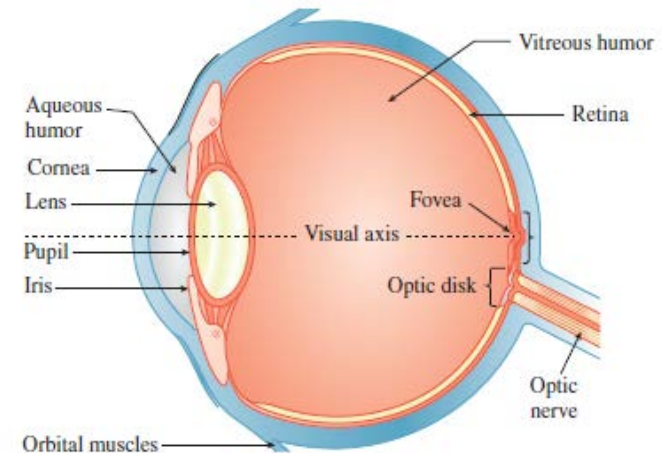
UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Modelo de cámara virtual

Visión 3D

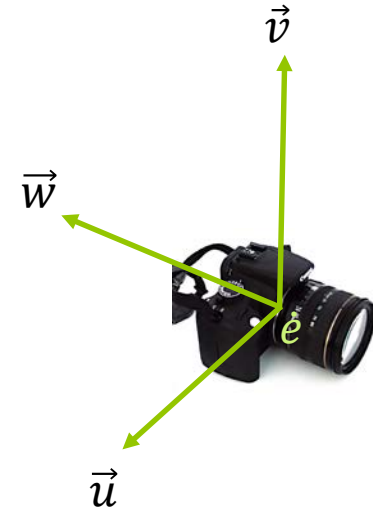
- ▶ **Visión 3D real**
 - ▶ Visión estéreo
 - ▶ Convergencia visual
 - ▶ Enfoque
 - ▶ Campo de visión
 - ▶ Perspectiva
 - ▶ Sensibilidad
- ▶ **Visión 3D simulada**
 - ▶ Intenta simular la visión real mediante el símil de una cámara fotográfica
 - ▶ Campo de visión: volumen de la vista
 - ▶ Estereoscopia: 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

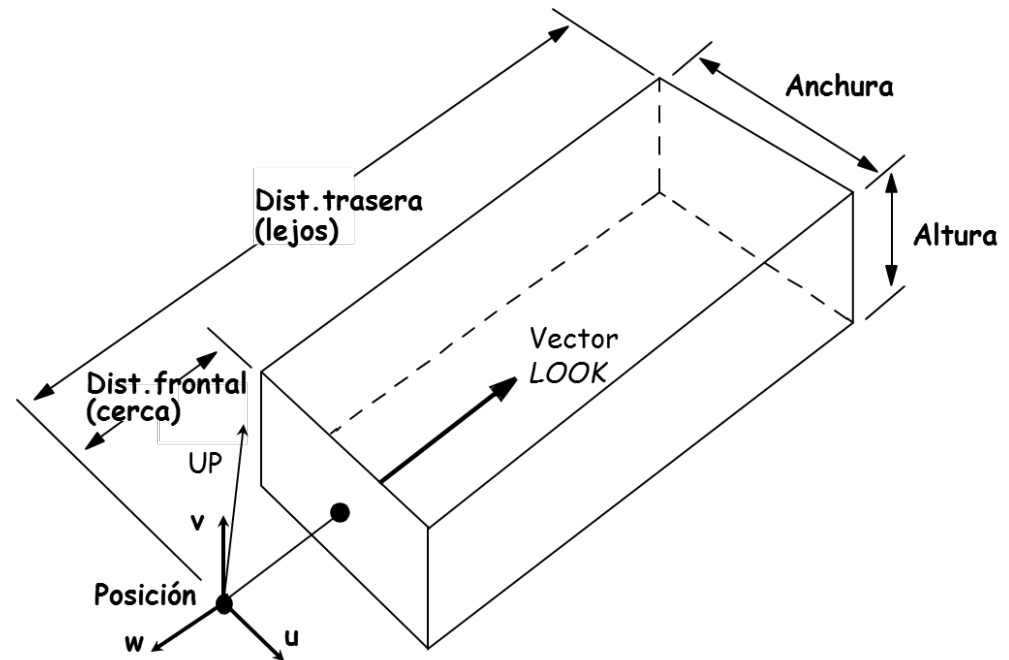
$$\vec{e}^T = [\vec{u} \quad \vec{v} \quad \vec{w} \quad e]$$



Cámara ortográfica

▶ Parámetros del frustum

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



*planos
delimitadores*

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

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

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

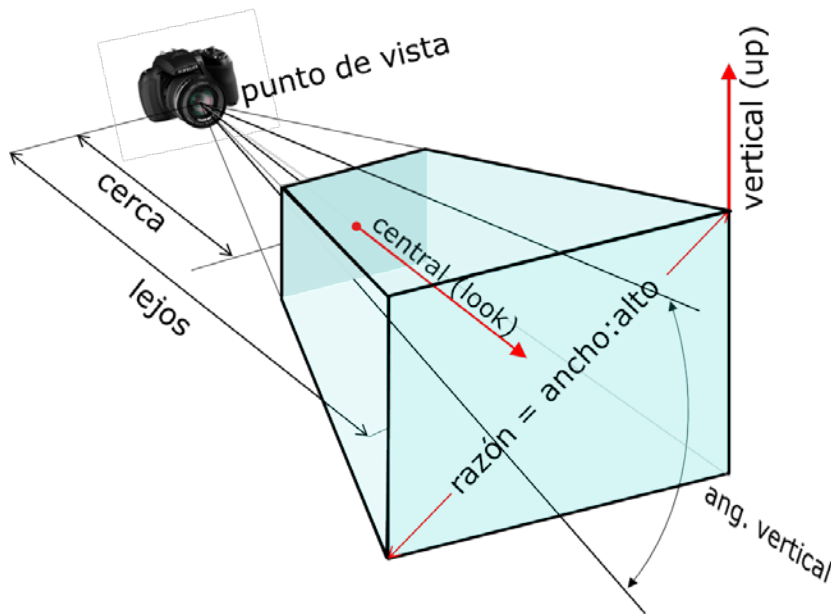
constructor

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

$$b = -t$$

$$r = ar \cdot t$$

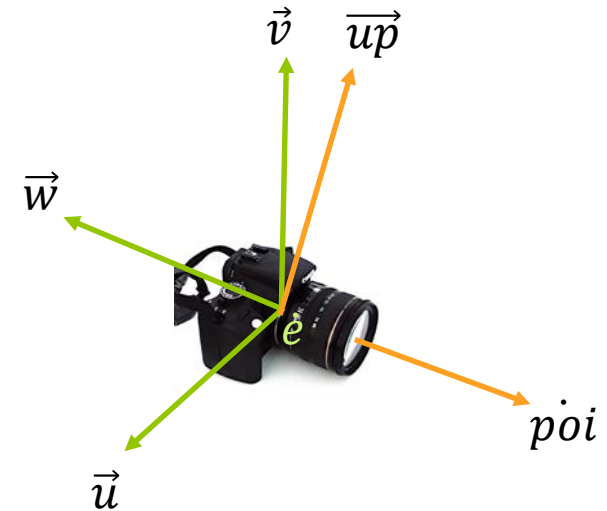
$$l = -r$$

$$n = -cerca$$

$$f = -lejos$$

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



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

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

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

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

método

```
camara.lookAt( e_x, e_y, e_z
               poi_x, poi_y, poi_z,
               up_x, up_y, up_z );
```



Sección de
Informática
Gráfica
VALENCIA

Computer
Graphics
Group



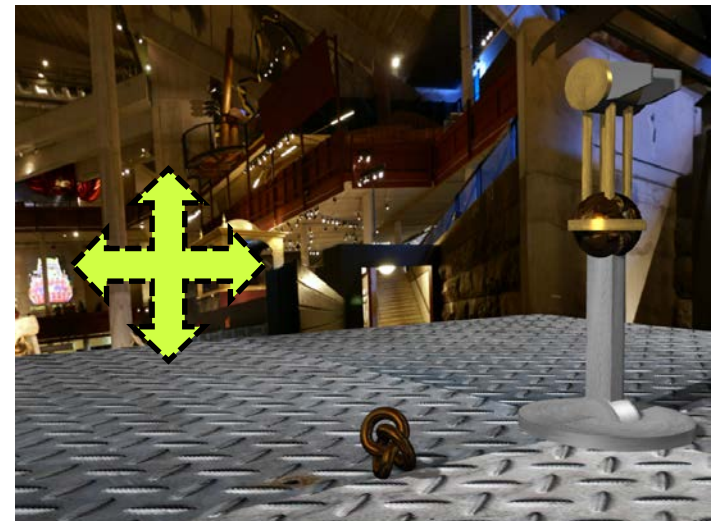
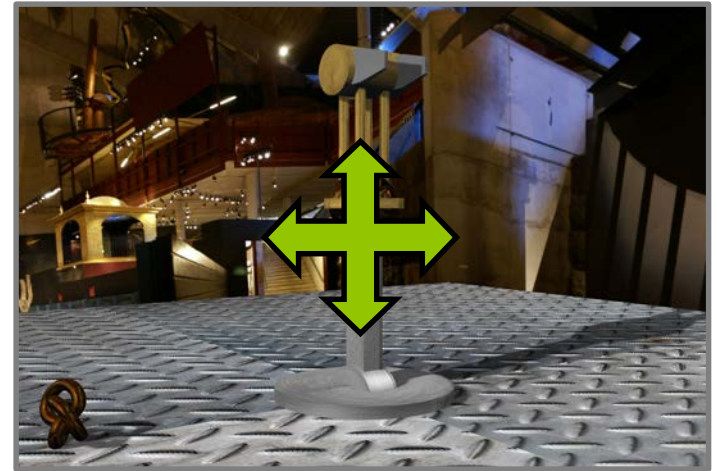
UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Interacción con la cámara

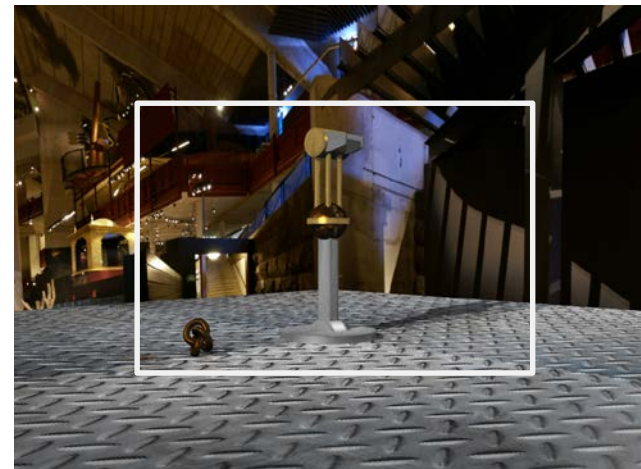
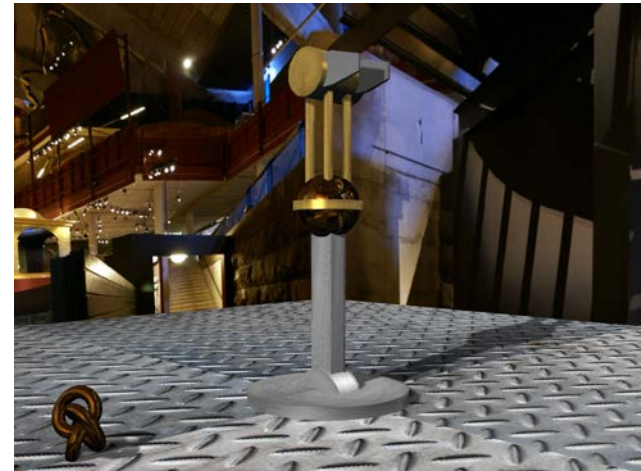
Desplazamiento (panning)

- ▶ 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 (*eje*) mediante una traslación en \vec{u}, \vec{v}
 - ▶ El punto de interés (*poi*)



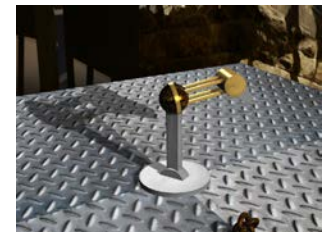
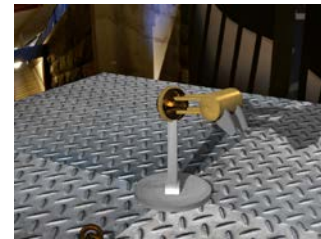
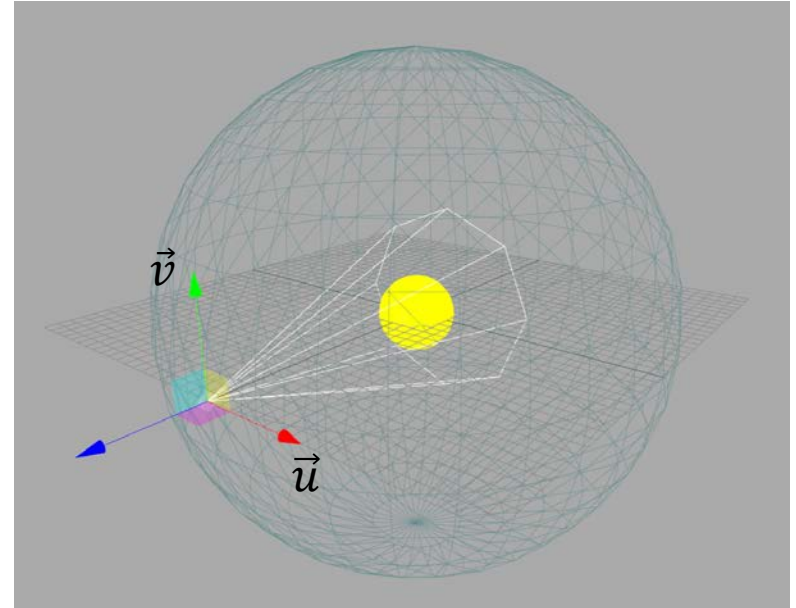
Acercar/Alejar (zoom)

- ▶ 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



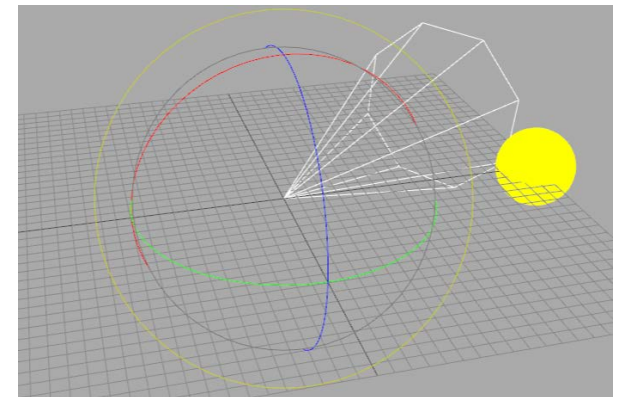
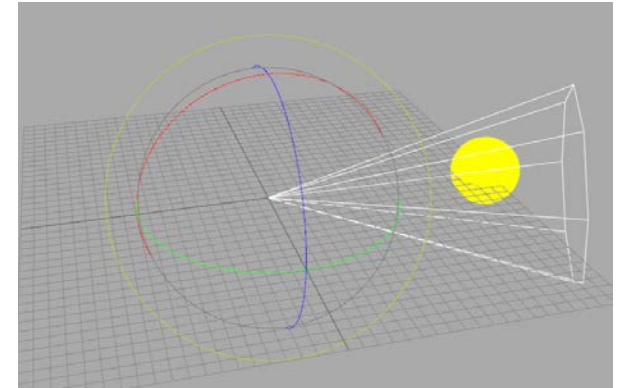
Orbital

- ▶ 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





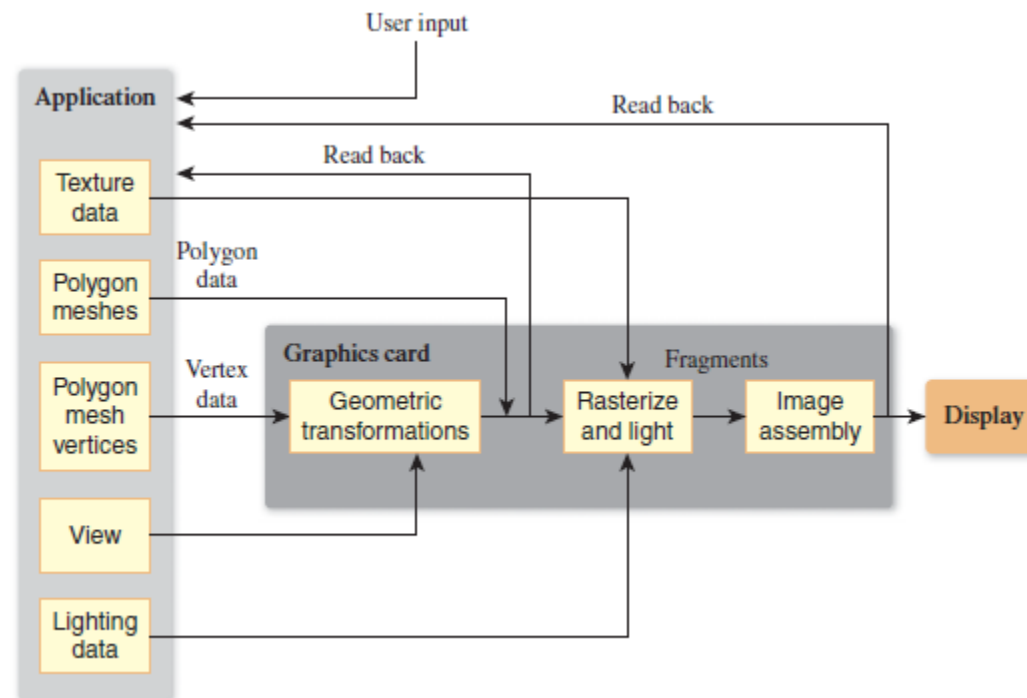
Rasterizing

Algoritmo básico

```
inicializar el z-buffer de lejanías
inicializar el buffer de colores (raster)
para cada triángulo
    proyectar sus vértices
    para cada píxel interior a la proyección
        calcular el color y la lejanía
        si está más cerca que el actual en el z-buffer
            actualizar el color en el raster y la lejanía en el z-buffer
```

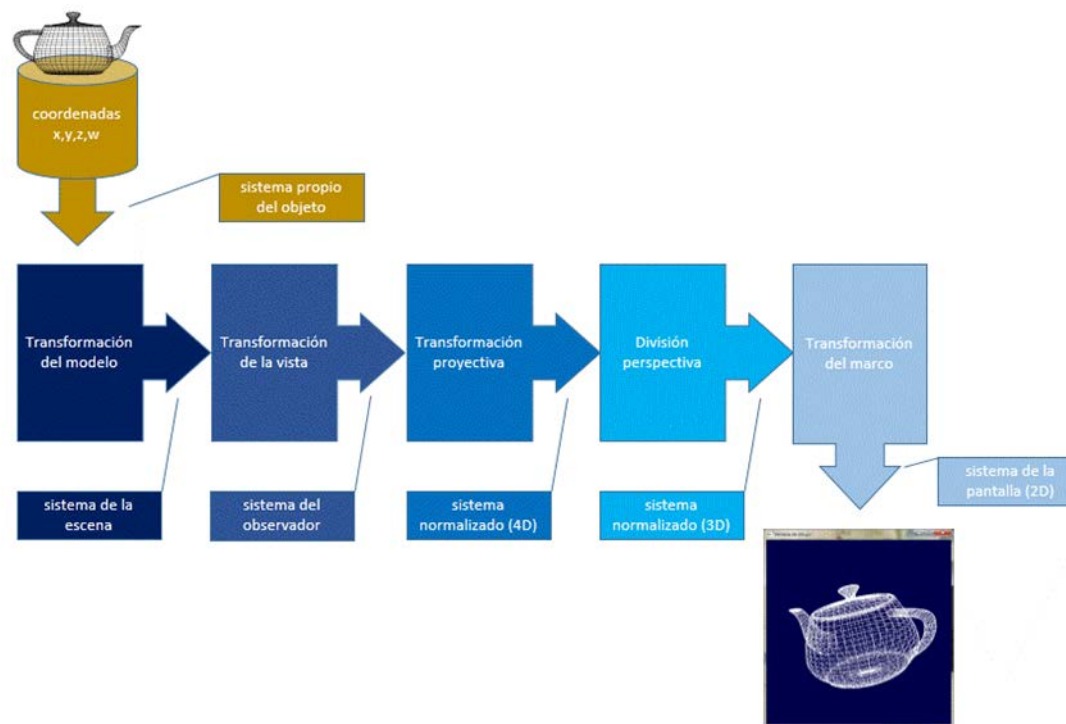
Tubería gráfica

- Secuencia de operaciones que convierten un objeto gráfico en píxeles

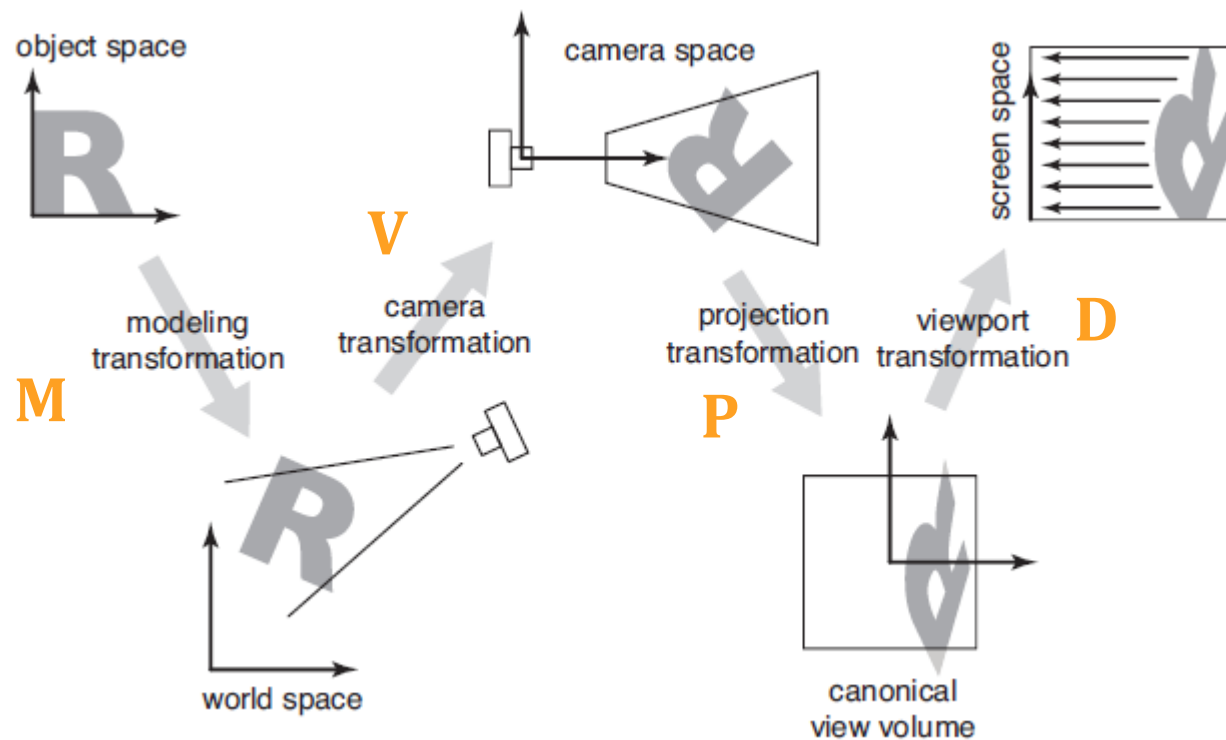


Procesado de vértices

► Transformaciones entre sistemas de referencia



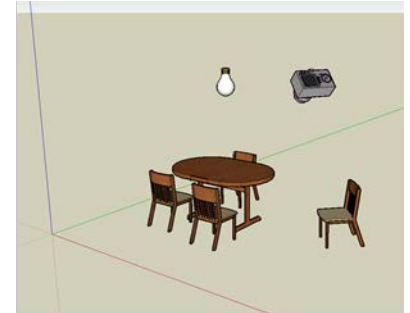
Cadena de transformaciones M-V-P-D



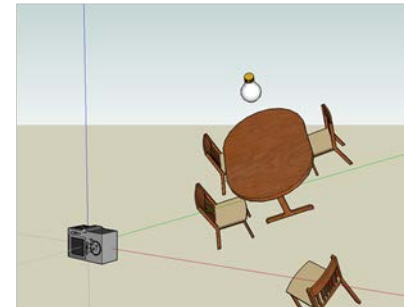
Trasformación de la vista (V)

- ▶ 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



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

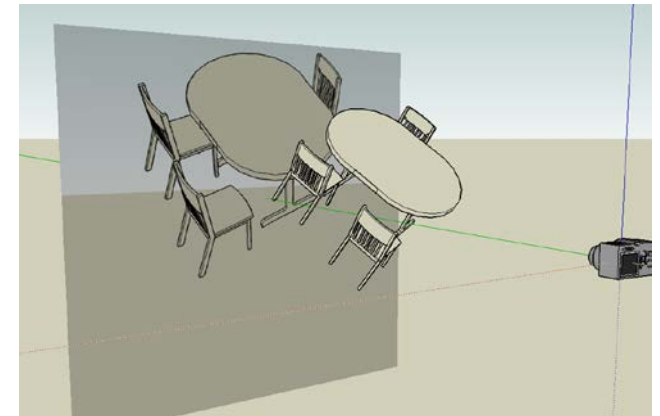
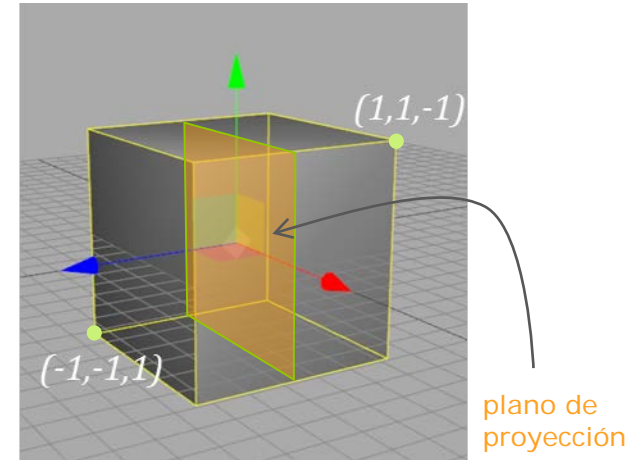
giro para bases
coincidentes

traslación para
orígenes coincidentes

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

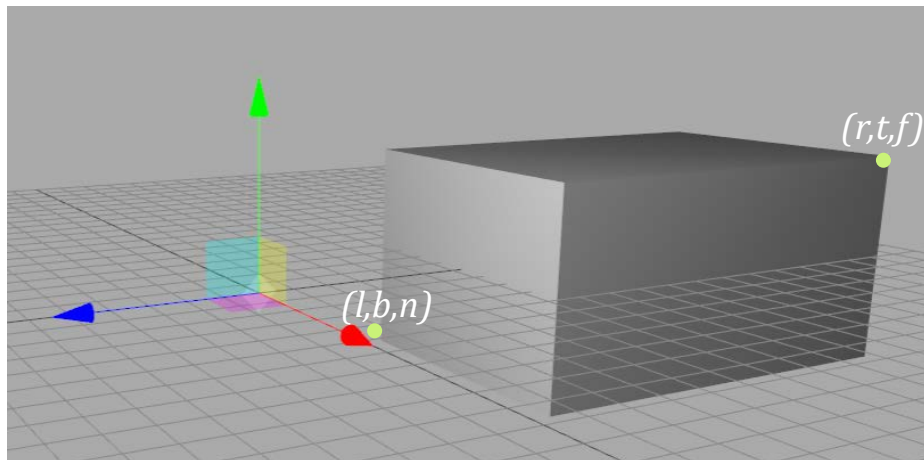
Trasformación proyectiva

- ▶ El sistema canónico de la proyección es una vista con un frustum cúbico de lado 2 centrado en el origen
- ▶ Dadas las coordenadas de un punto en nuestro sistema de la vista queremos conocer sus coordenadas en el sistema canónico de la proyección (3D)
- ▶ La ventana canónica de la proyección (2D) es un cuadrado de 2x2 centrado en el origen
- ▶ En la proyección ortográfica, las coordenadas del punto proyectado sobre la ventana canónica son las coordenadas x,y del punto sobre el sistema canónico de la proyección
- ▶ Podemos convertir un frustum piramidal (cámara perspectiva) en uno ortoédrico deformando el espacio. Así las coordenadas del punto proyectado se calcularan como si la proyección fuera ortográfica

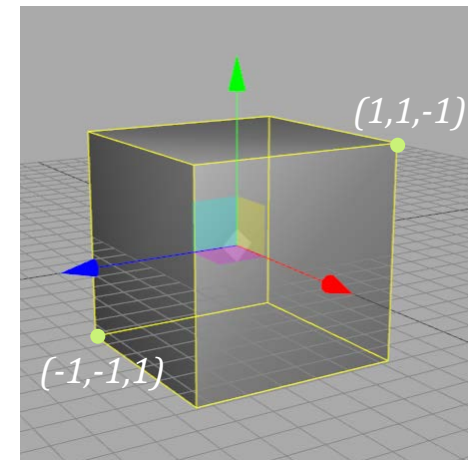


Trasformación proyectiva ortográfica

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



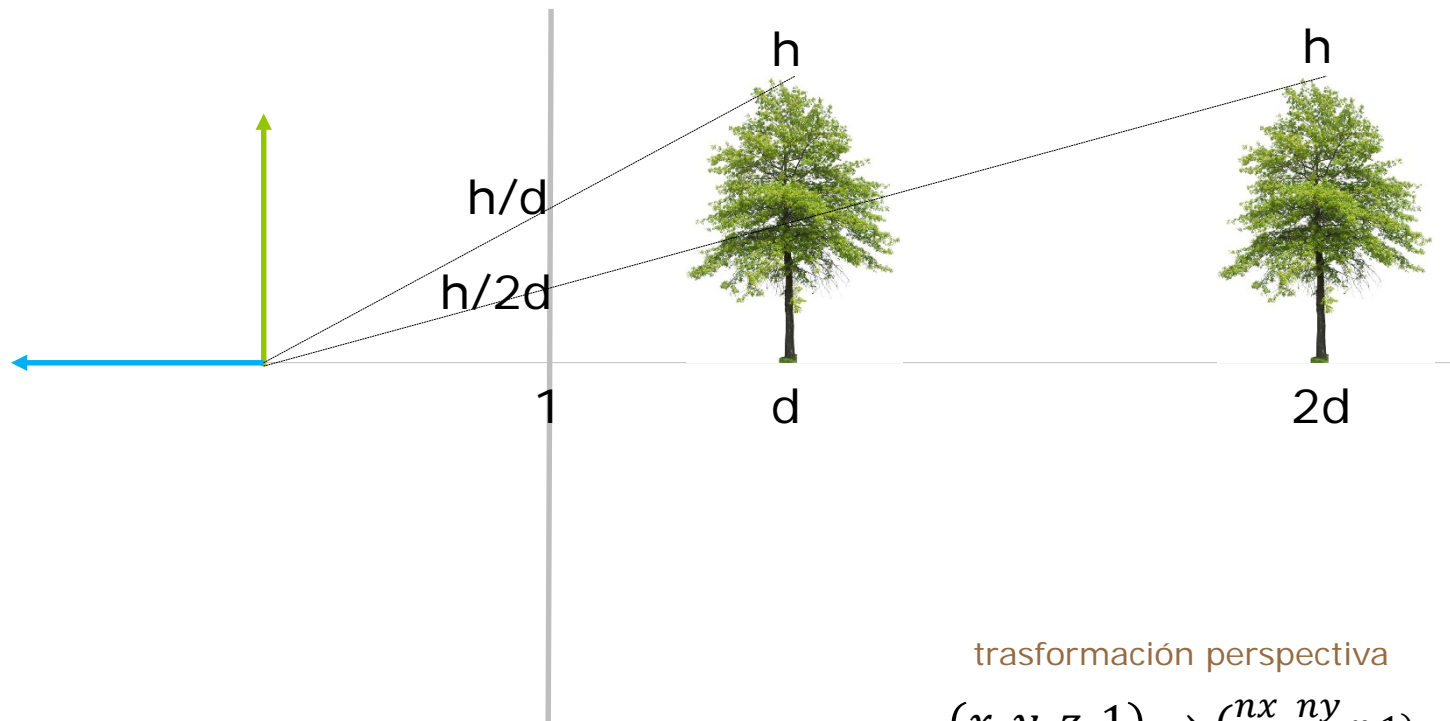
frustum ortográfico



frustum canónico

Proyección perspectiva

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

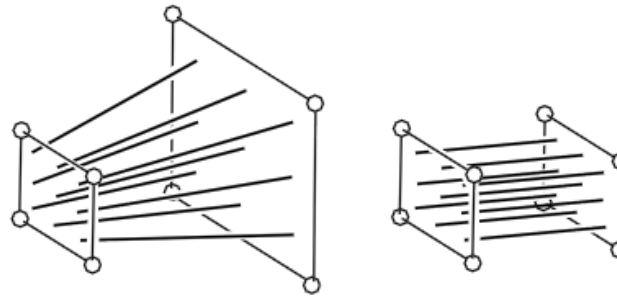


transformación perspectiva

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

Trasformación proyectiva perspectiva

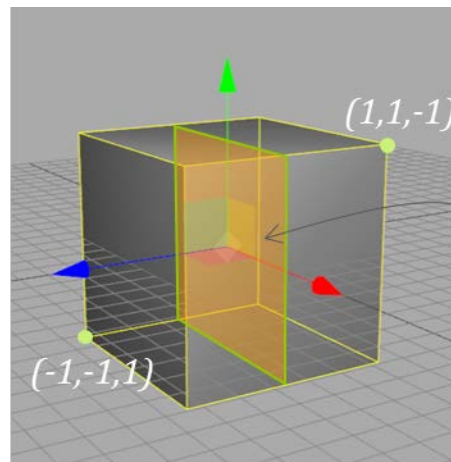
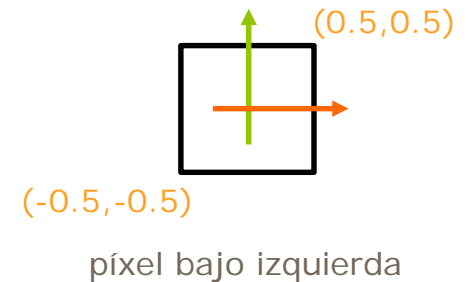
1. Convertimos el volumen de la vista perspectiva en un ortoedro deformando los objetos



- 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

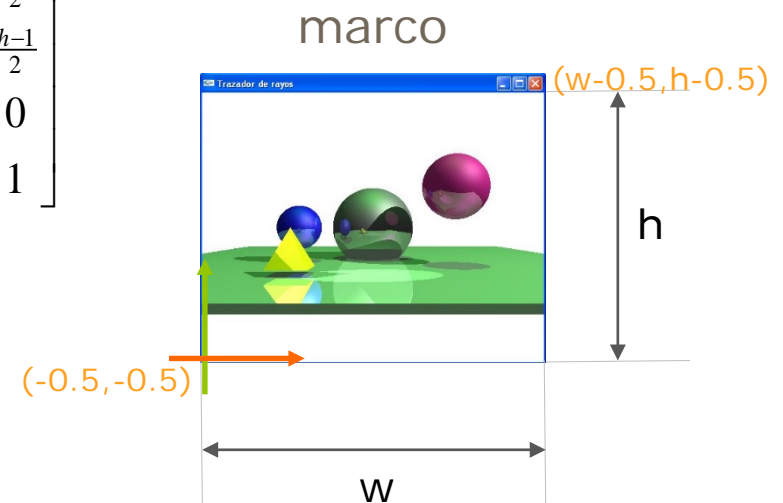
Trasformación del marco

- ▶ 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)

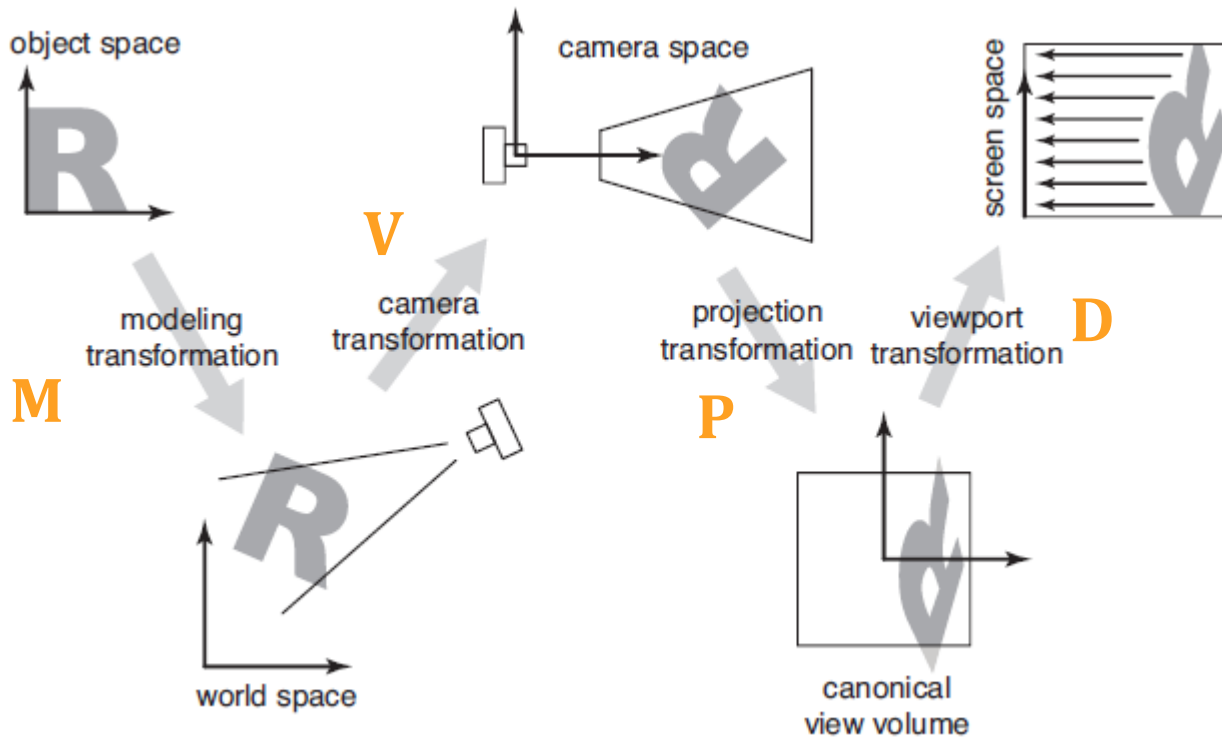


plano de
proyección
2x2

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



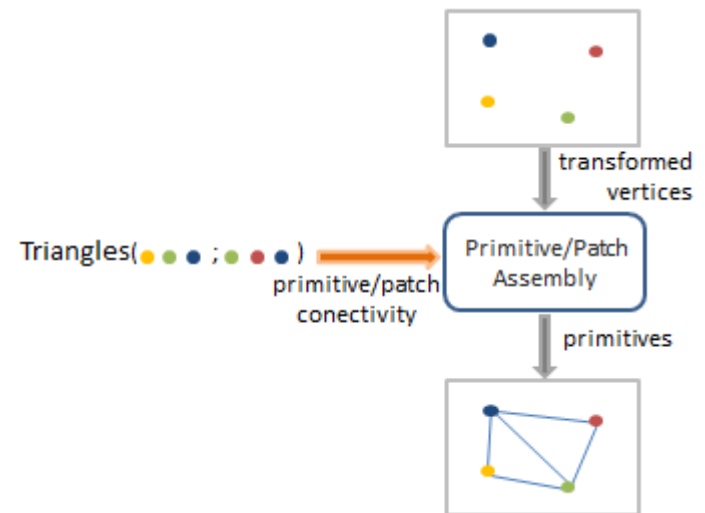
Cadena de transformaciones M-V-P-D



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

Ensamblado de primitivas

- ▶ Proceso de unión de los vértices para formar triángulos
- ▶ Debe existir información sobre cómo se enlazan los vértices (conectividad)
 - ▶ Explícita: Índices a los vértices
 - ▶ Implícita: El orden de procesamiento
- ▶ La información de conectividad no varía con las transformaciones geométricas (procesado de vértices)
- ▶ El orden de enlace de los vértices determina la cara, exterior o interior, del triángulo



Recortado

- ▶ Proceso por el que se eliminan los triángulos o la parte de ellos que está fuera del frustum
- ▶ Se realiza en el sistema normalizado 4D (cubo canónico)
- ▶ El polígono resultante es convexo (máximo 6 lados)
- ▶ Los atributos de los nuevos vértices se interpolan en la arista

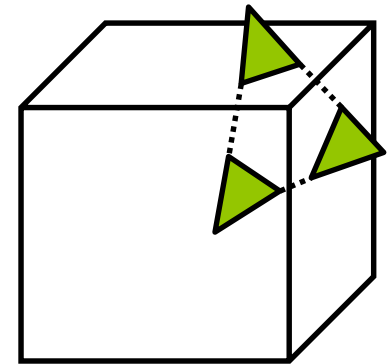
condición
de interioridad

$$p(x, y, z, w)$$

$$-w < x < w$$

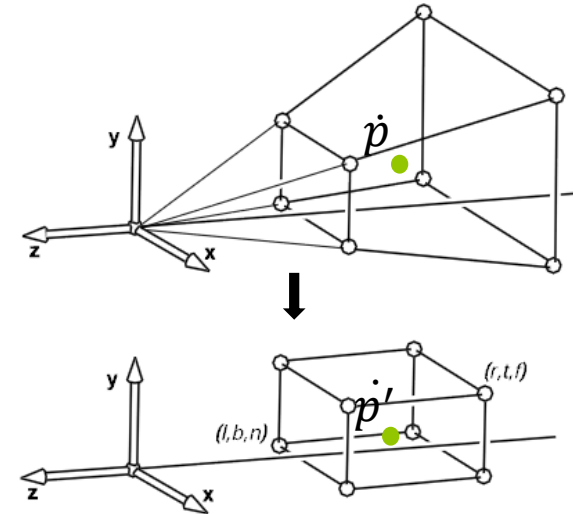
$$-w < y < w$$

$$-w < z < w$$



División perspectiva

- ▶ La matriz de proyección perspectiva deforma la pirámide hasta una caja
- ▶ Supone el paso a un sistema 4D
- ▶ Para volver al sistema 3D es necesario dividir por la cuarta coordenada w



$$\dot{p} = [p_x \ p_y \ p_z \ 1]^T$$

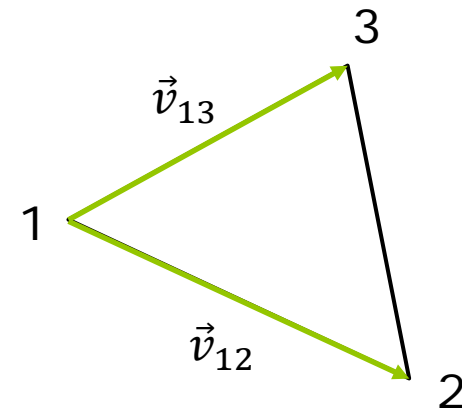
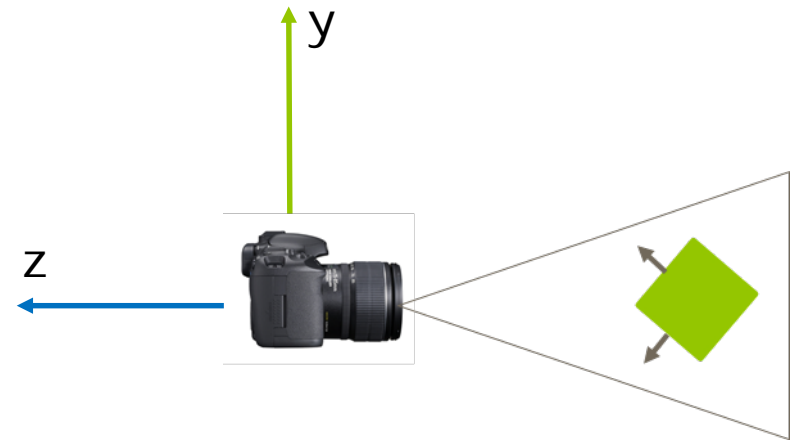
$$\mathbf{M}_p = \begin{bmatrix} \mathbf{S} & \mathbf{T} \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\dot{q} = \mathbf{M}_p \dot{p}$$

$$\dot{p}' = \dot{q} \div q_w$$

Eliminación de caras

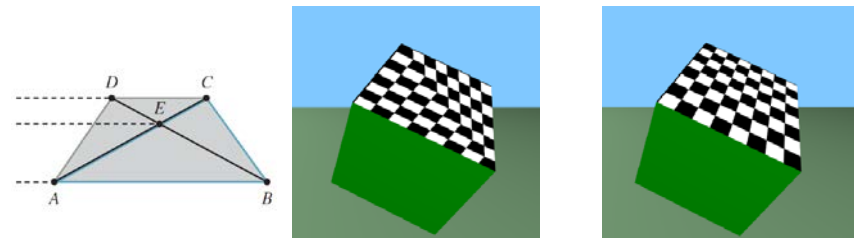
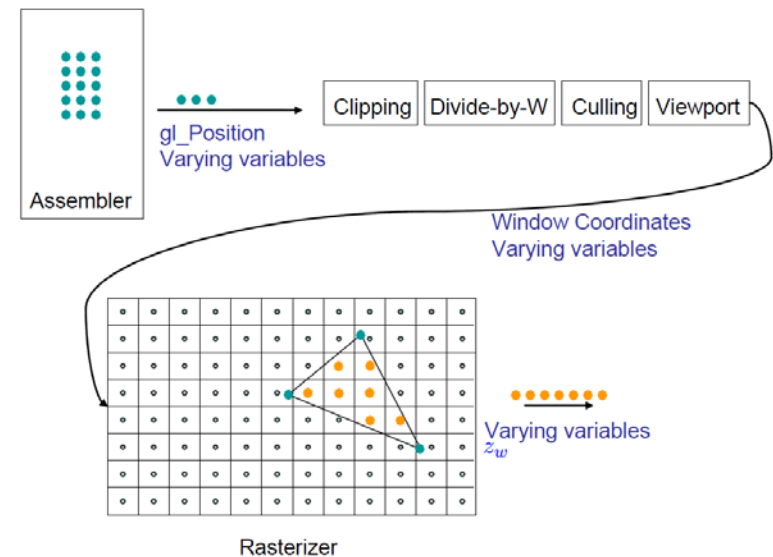
- ▶ En objetos cerrados las caras traseras no se ven
- ▶ Una cara es trasera si la componente z en el sistema de la cámara de su normal es negativa
- ▶ Establecido un criterio de ordenación: simple producto vectorial
- ▶ Se suele calcular en el sistema normalizado 3D



$$\text{signo}((v_{12x}v_{13y}) - (v_{12y}v_{13x}))$$

Muestreo e interpolación

- ▶ “Rasterización”: muestreo para determinar píxeles interiores a la forma en el sistema del marco
- ▶ Rasterización
 - ▶ de puntos
 - ▶ de líneas
 - ▶ de triángulos
- ▶ Los valores de los atributos en los píxeles se interpolan a partir de los de los vértices
- ▶ La interpolación debe tener en cuenta la distorsión perspectiva



$$f(e) = \frac{\frac{\alpha}{a_z} f(a) + \frac{\beta}{c_z} f(c) + \frac{\gamma}{d_z} f(d)}{\frac{\alpha}{a_z} + \frac{\beta}{c_z} + \frac{\gamma}{d_z}}$$

α, β, γ
coordenadas
baricéntricas en
la proyección

Procesado de fragmentos

- ▶ Los fragmentos son el resultado de la etapa de discretización pudiendo haber varios para el mismo píxel
- ▶ Cada fragmento se caracteriza por:
 - ▶ Posición i, j en el raster
 - ▶ Profundidad z normalizada
 - ▶ Color
 - ▶ Otros atributos como
 - ▶ Transparencia
 - ▶ Normal
 - ▶ Coordenadas de texturas
 - ▶ Tangente
 - ▶ ...
- ▶ Procesar un fragmento supone
 - ▶ Aceptarlo / Descartarlo
 - ▶ Calcular su color
 - ▶ Posiblemente, modificar el color del píxel correspondiente

Blending y z-buffer

▶ Z-Buffer

- ▶ Técnica por la que se acepta o descarta un fragmento según su cercanía comparada con la del píxel actual
- ▶ Es el método de resolución del problema de visibilidad
- ▶ El z-buffer puede desactivarse en escritura de tal manera que un fragmento no actualice la cercanía del píxel
 - ▶ Ordenación traslúcidos y opacos

▶ Blending

- ▶ Técnica por la cuál se mezcla el color del fragmento con el color actual del píxel
- ▶ Mediante “blending” se consiguen objetos traslúcidos, antialiasing, ...
- ▶ Viene determinada por una función del RGBA del fragmento y el RGBA actual del píxel

Escritura en el buffer

- ▶ Una vez procesados todos los fragmentos el buffer de color está actualizado
- ▶ El buffer de color puede usarse para:
 - ▶ Actualizar el framebuffer del dispositivo
 - ▶ frontbuffer, backbuffer
 - ▶ Construir una textura que sirve de entrada al siguiente render
 - ▶ Mapas de entorno
 - ▶ Mapas de sombras
 - ▶ Acumularlo
 - ▶ Antialiasing



Sección de
Informática
Gráfica | Computer
Graphics
Group
VALENCIA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



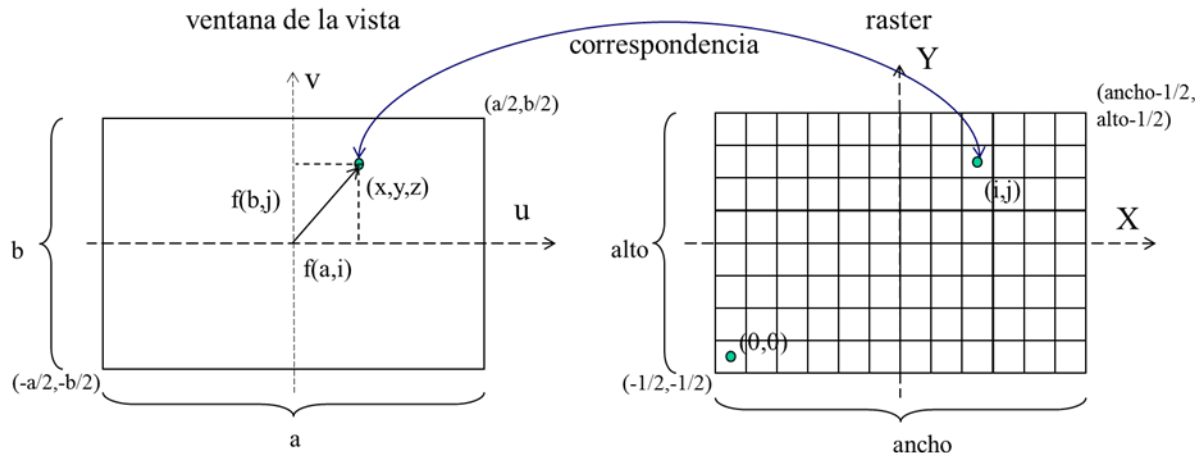
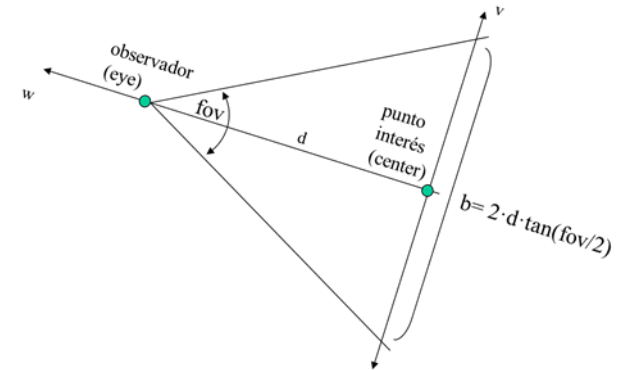
Trazado de rayos

Algoritmo básico

```
inicializar el color en el raster
para cada píxel del raster
    calcular la visual que pasa por su centro
    para cada triangulo
        calcular la intersección con la visual
    si hay intersección
        elegir la intersección más cercana al observador
        calcular el color en ese punto
        actualizar el color del pixel
```

Fragmentos y visuales

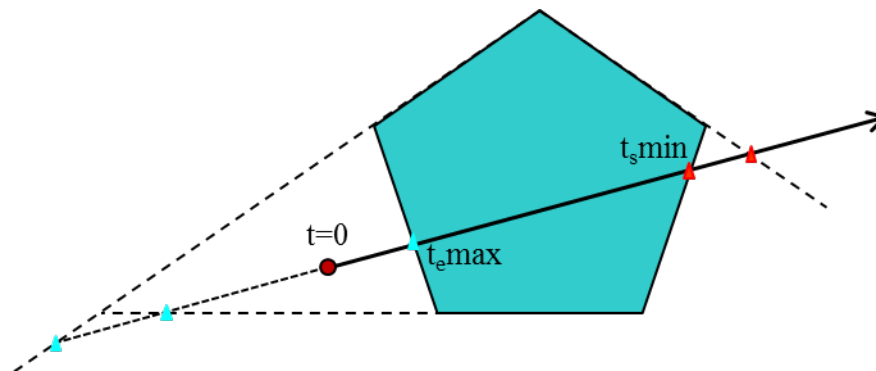
- ▶ Una visual es una semirrecta (rayo) que parte del ojo y pasa por un punto de un fragmento
 - ▶ punto de origen
 - ▶ vector dirección
- ▶ Cálculo de la visual
 - ▶ Fijar las dimensiones del ráster
 - ▶ Definir la cámara con la misma razón de aspecto
 - ▶ frustum
 - ▶ sistema de referencia u, v, w
 - ▶ Elegir el punto de paso (habitualmente el centro del píxel)
 - ▶ Calcular las coordenadas del punto de paso en la ventana de proyección (x_{win}, y_{win})



origen: ojo
dirección: unitario($-\vec{w} + x_{win}\vec{u} + y_{win}\vec{v}$)

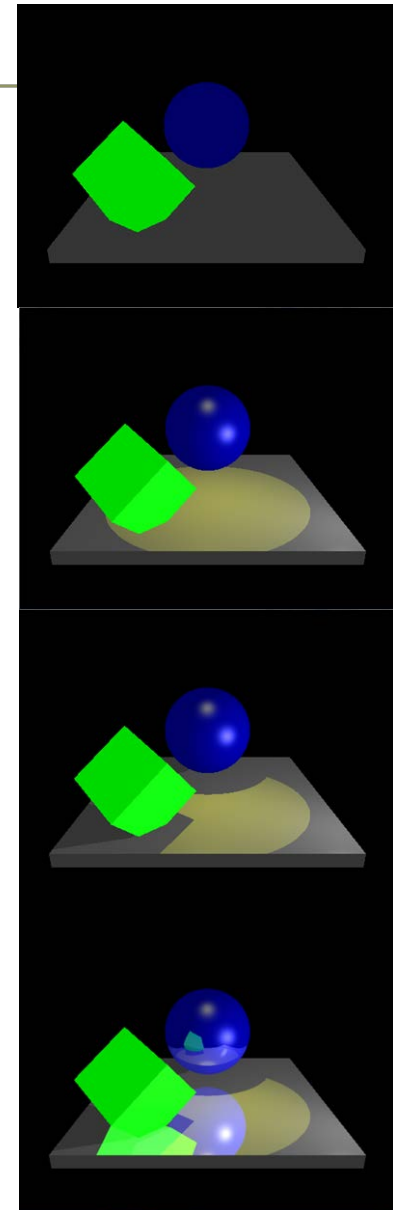
Trazado de un rayo

- ▶ Trazar un rayo significa calcular las intersecciones con todos los objetos de la escena
 - ▶ intersección rayo-polígono
 - ▶ intersección rayo-poliedro convexo
 - ▶ intersección rayo-cuádrica
- ▶ La intersección más cercana al ojo indica el punto visible
- ▶ Métodos de aceleración por contenedores



Cálculo del color

- ▶ Se aplica un modelo de iluminación en el punto intersección que contabiliza
 - ▶ iluminación directa
 - ▶ modelo de Blinn-Phong con oclusión
 - ▶ reflexión indirecta
 - ▶ modelo de Whitted
- ▶ Para el cálculo del color es importante
 - ▶ Trazar rayos a las fuentes de luz
 - ▶ Trazar rayos secundarios de reflexión y transmisión perfecta (árbol de rayos)



Escritura en el buffer

- ▶ Conforme se calcula el color se va rellenando el buffer
 - ▶ Si la visual no intersecó: color de fondo
 - ▶ Si hubo intersección: color calculado
- ▶ Problema de saturación
 - ▶ Remuestreo
- ▶ Problema de Aliasing: trazar varias visuales por fragmento
 - ▶ Antialiasing uniforme
 - ▶ Antialiasing adaptativo
 - ▶ Jittering

