



Sección de
Informática
Gráfica | Computer
Graphics
Group
VALENCIA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Rendering: Polígonos vs. Rayos

Rendering

Procesado de polígonos

Procesado de rayos



Sección de
Informática
Gráfica | Computer
Graphics
Group
VALENCIA



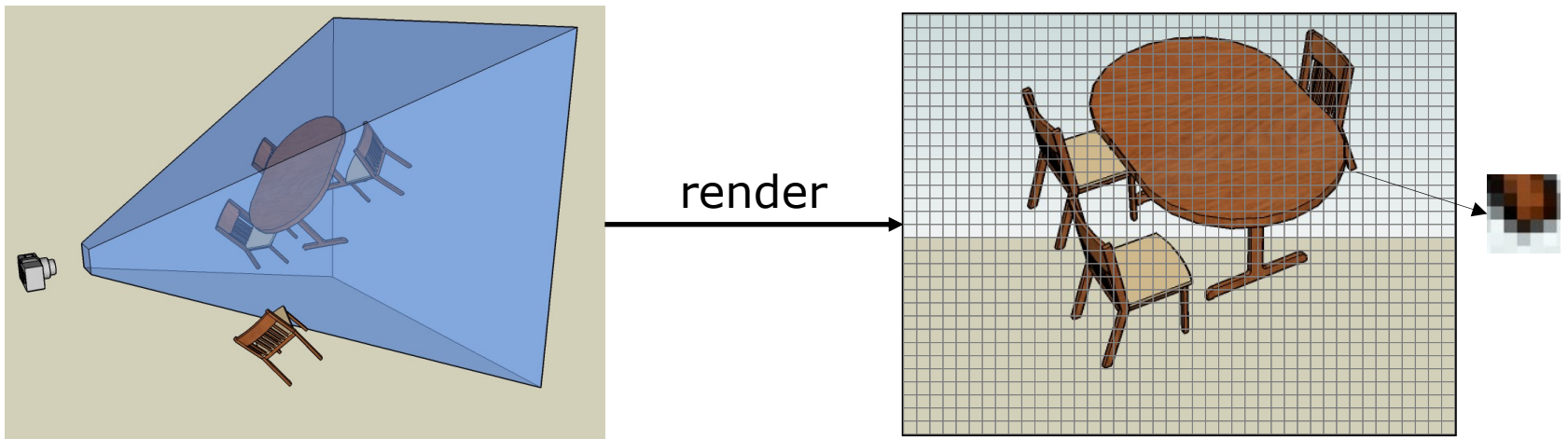
UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



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

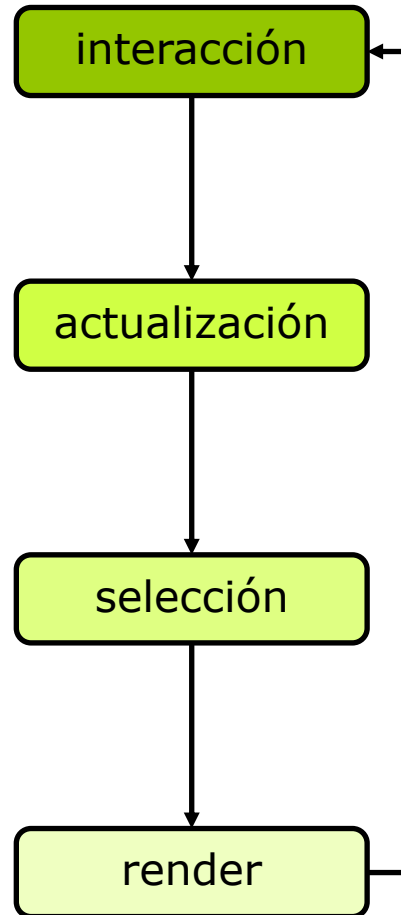
Bucle de animación

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

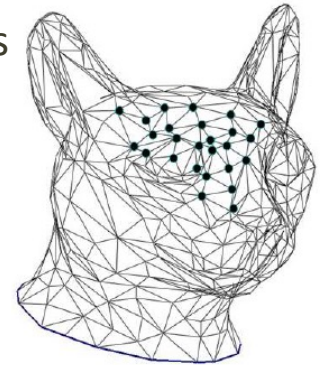
Se convierte la
información en imagen



Protagonistas del render

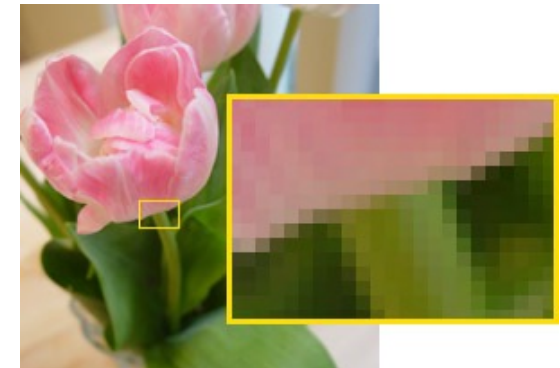
▶ Vértices

- ▶ 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



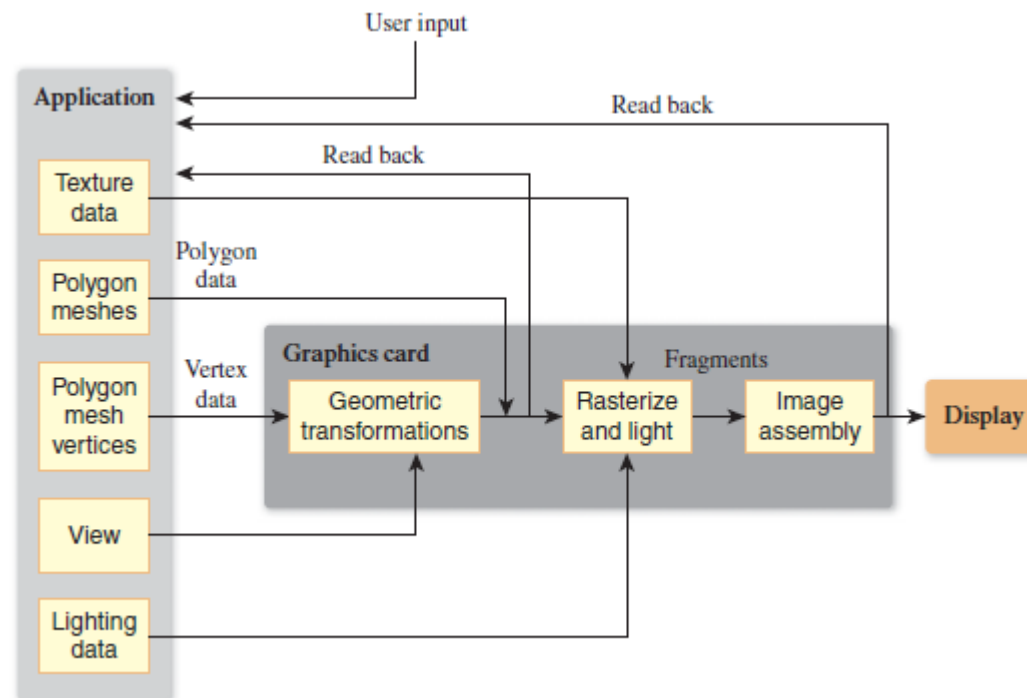
Rasterizing

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
```

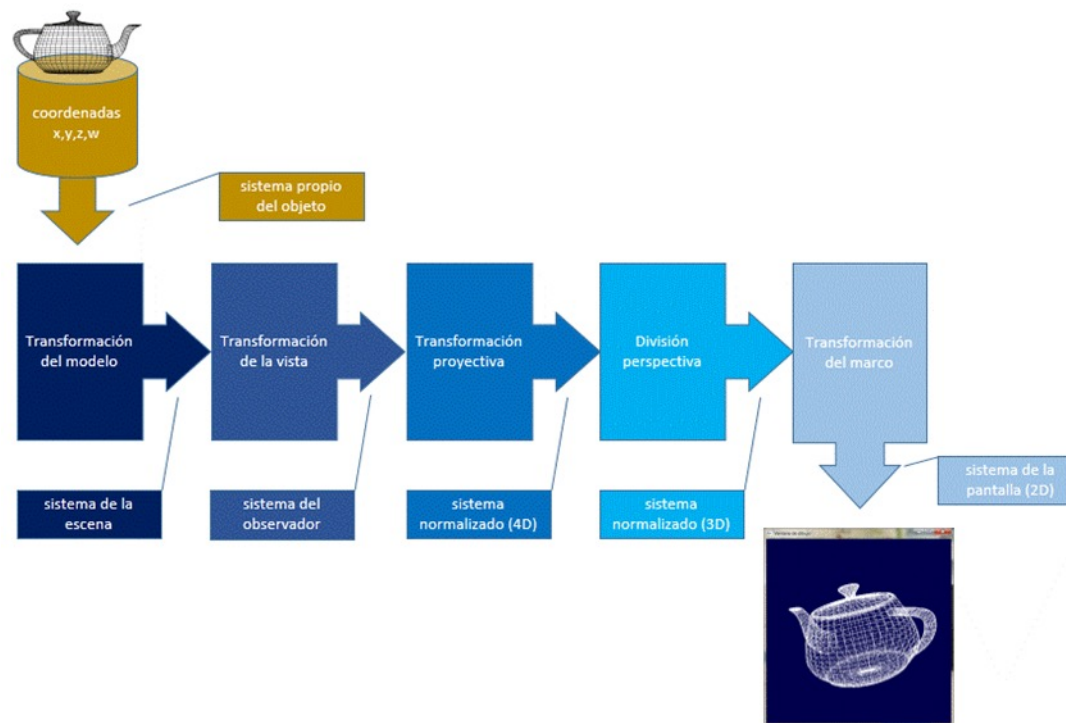
Tubería gráfica

- Secuencia de operaciones que convierten un objeto gráfico en píxeles



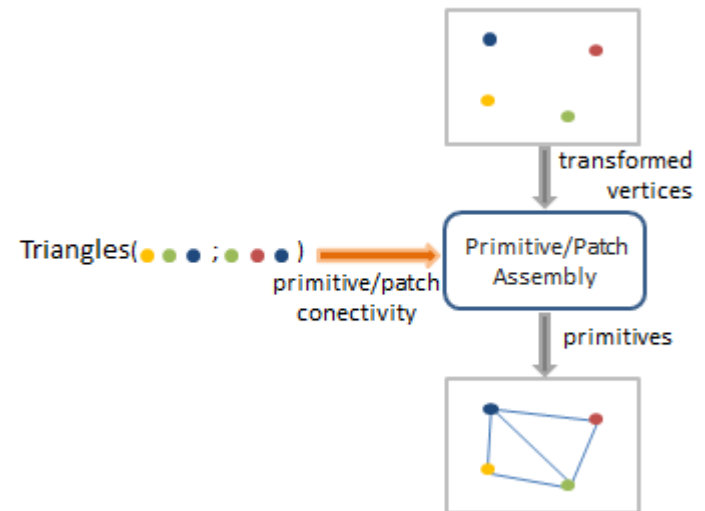
Procesado de vértices

► Transformaciones entre sistemas de referencia



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

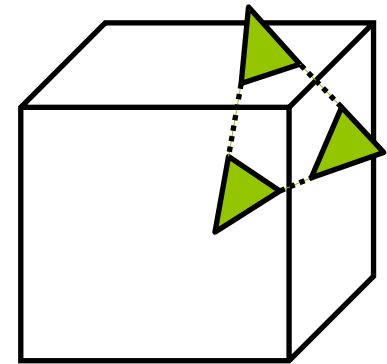


Recortado

- ▶ 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

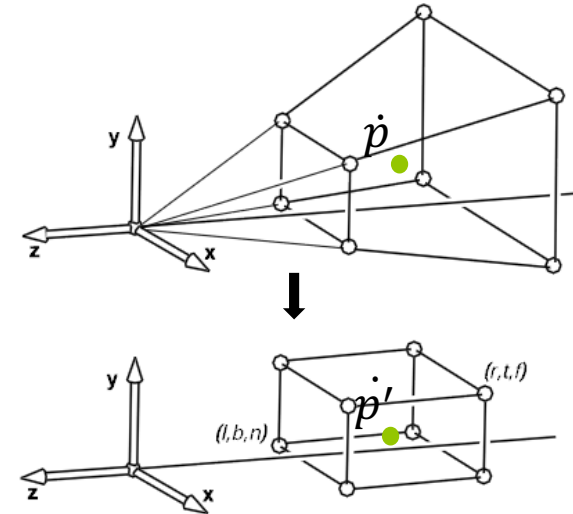
condición
de interioridad

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



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



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

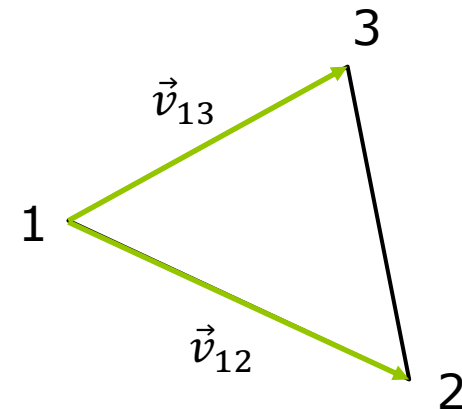
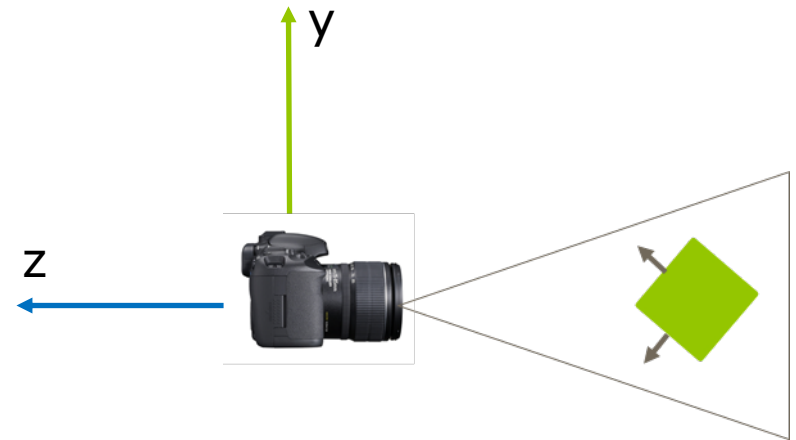
$$\mathbf{M}_p = \begin{bmatrix} \mathbf{S} & \mathbf{T} \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

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

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

Eliminación de caras

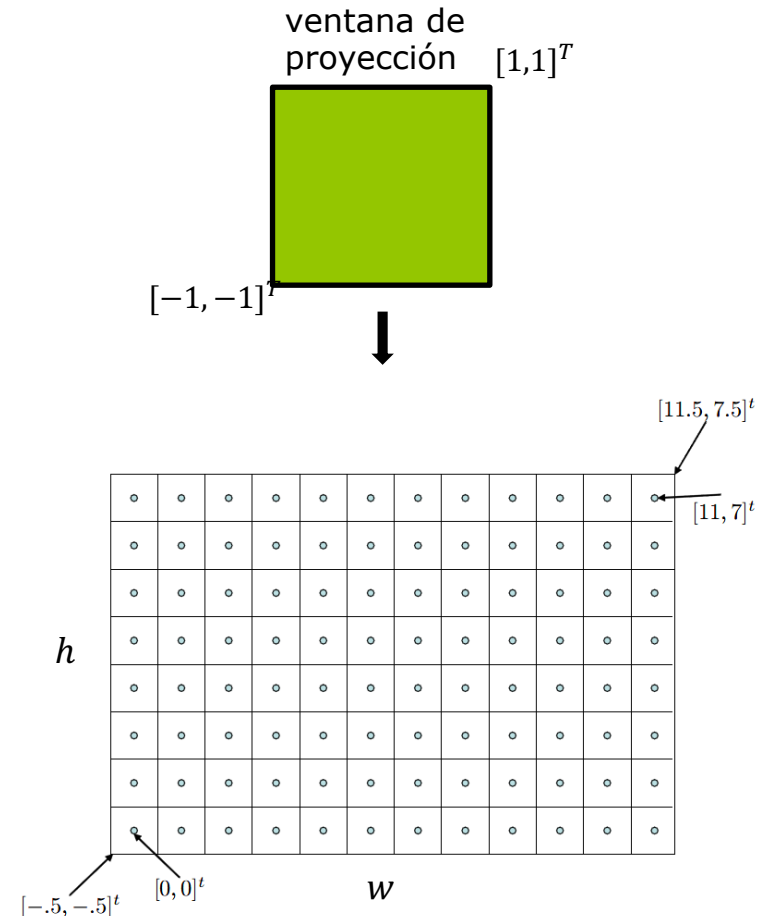
- ▶ En objetos cerrados las caras traseras no se ven
- ▶ Una cara es trasera si la componente z en el sistema de la cámara de su normal es negativa
- ▶ Establecido un criterio de ordenación: simple producto vectorial
- ▶ Se suele calcular en el sistema normalizado 3D



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

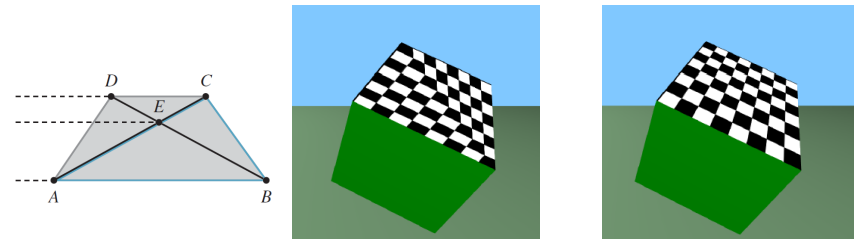
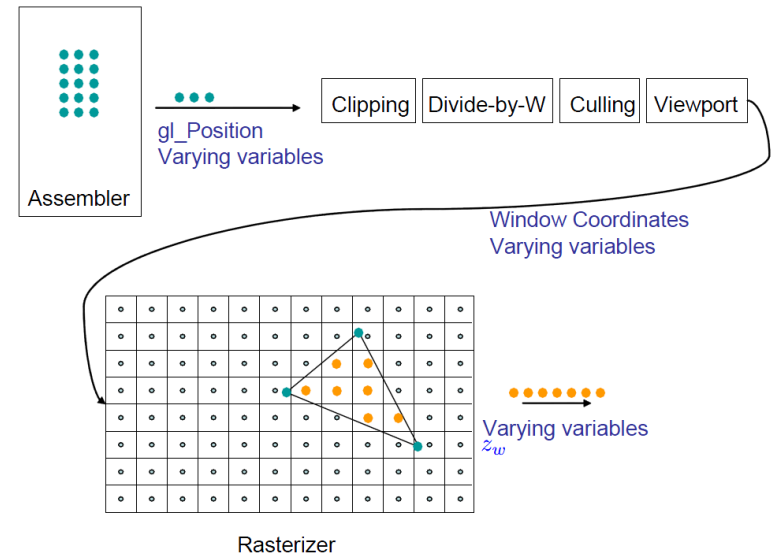
Marco (viewport)

- ▶ El marco es el rectángulo 2D donde se encaja la ventana de proyección normalizada
- ▶ El marco se caracteriza por el número de píxeles horizontales y verticales ($w \times h$)
- ▶ La transformación del marco consiste en un escalado y una traslación en 2D



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



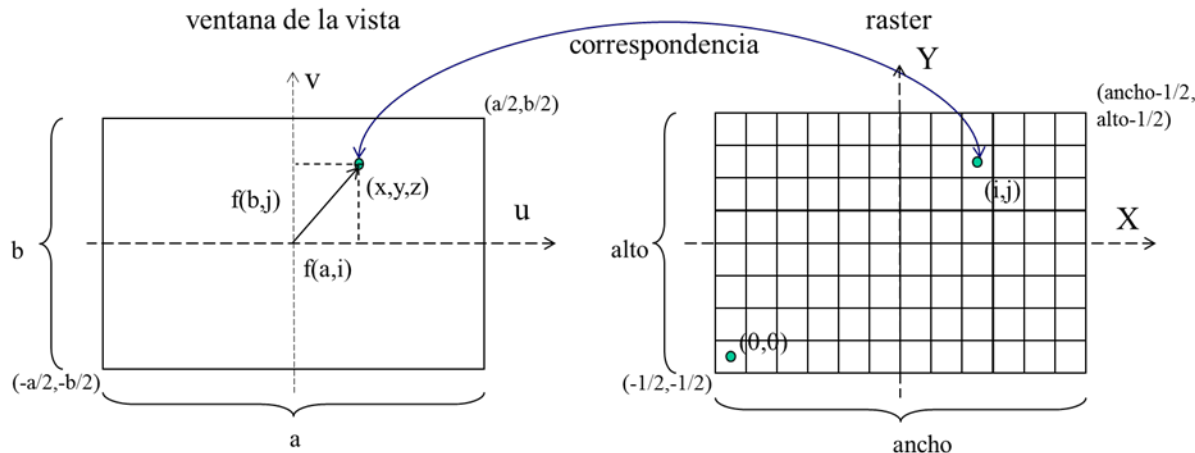
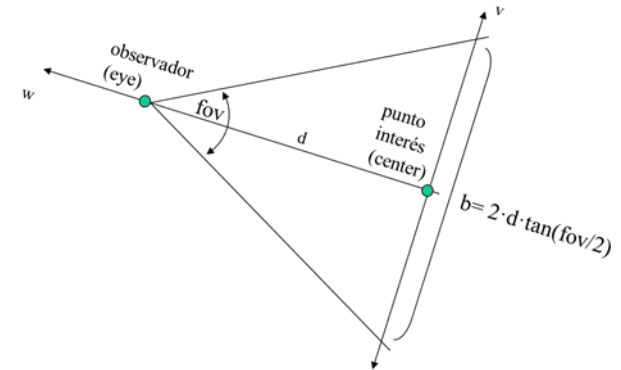
Trazado de rayos

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

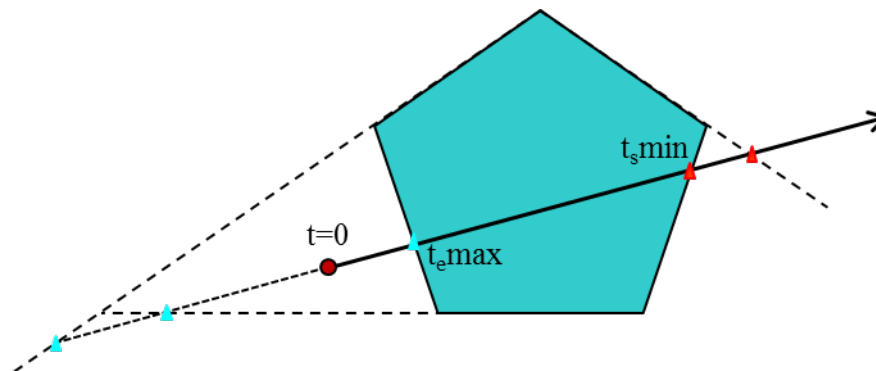
- ▶ Una visual es una semirrecta (rayo) que parte del ojo y pasa por un punto de un fragmento
 - ▶ punto de origen
 - ▶ vector dirección
- ▶ Cálculo de la visual
 - ▶ Fijar las dimensiones del ráster
 - ▶ Definir la cámara con la misma razón de aspecto
 - ▶ frustum
 - ▶ sistema de referencia u, v, w
 - ▶ 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($-\vec{w} + x_{win}\vec{u} + y_{win}\vec{v}$)

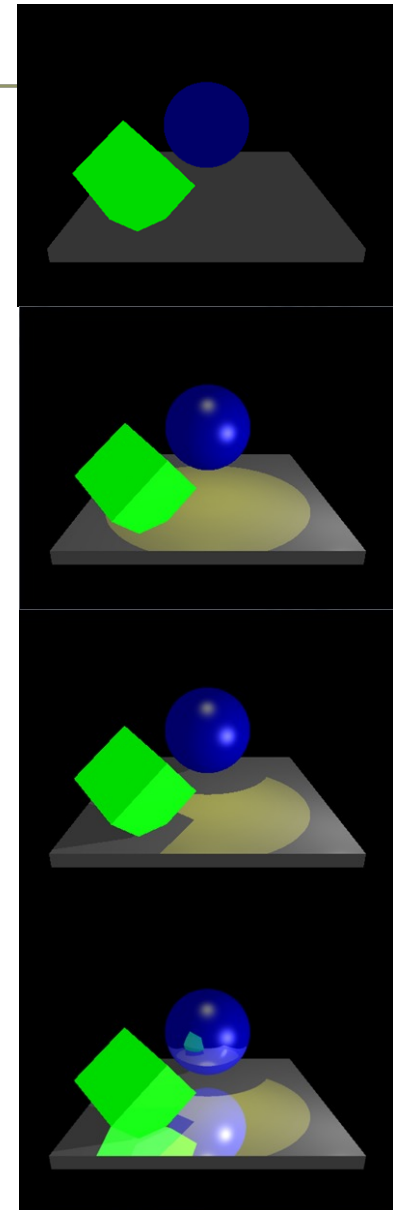
Trazado de un rayo

- ▶ 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

