

Actividad

Definición General: El ocultamiento de superficies es el proceso usado para determinar que superficies y partes de superficies son visibles o no desde algún punto de observación.

Principales Métodos:

1. **Eliminación de caras ocultas (back face culling) :** Consiste en eliminar los polígonos cuyas caras frontales no están frente a la cámara o en el punto de visión de las mismas.

Características:

- Esto se basa en clasificar las caras de los polígonos en traseras o delanteras. Aquellas caras traseras frente a la cámara se borran.
- Cuando los polígonos son triángulos es fácil determinar si su cara de enfrente está o no frente a la cámara. Un triángulo se elimina si el producto punto entre su normal y el vector de la cámara al triángulo es mayor o igual que cero, es decir, $(V_0 - P) \cdot N \geq 0$, donde, P es la posición de la cámara, V_0 es el primer vértice del triángulo, N es la normal ($N = (V_1 - V_0) \times (V_2 - V_0)$).

Ventajas:

- Se puede emplear este método antes del prewarping porque ahorra proceso de proyección.
-

Desventajas:

- Solo sirve para objetos modelados con una red (mesh) de polígonos.
- Funciona para objetos cóncavos, pero no necesariamente para convexos.

2. **Algoritmo del pintor:** Este algoritmo es básico y solo sigue dos pasos fundamentales:

- Ordenar los polígonos usando su coordenada z.
- Pintar los polígonos de atrás (z máxima) hacia delante (z mínima).

Ventajas:

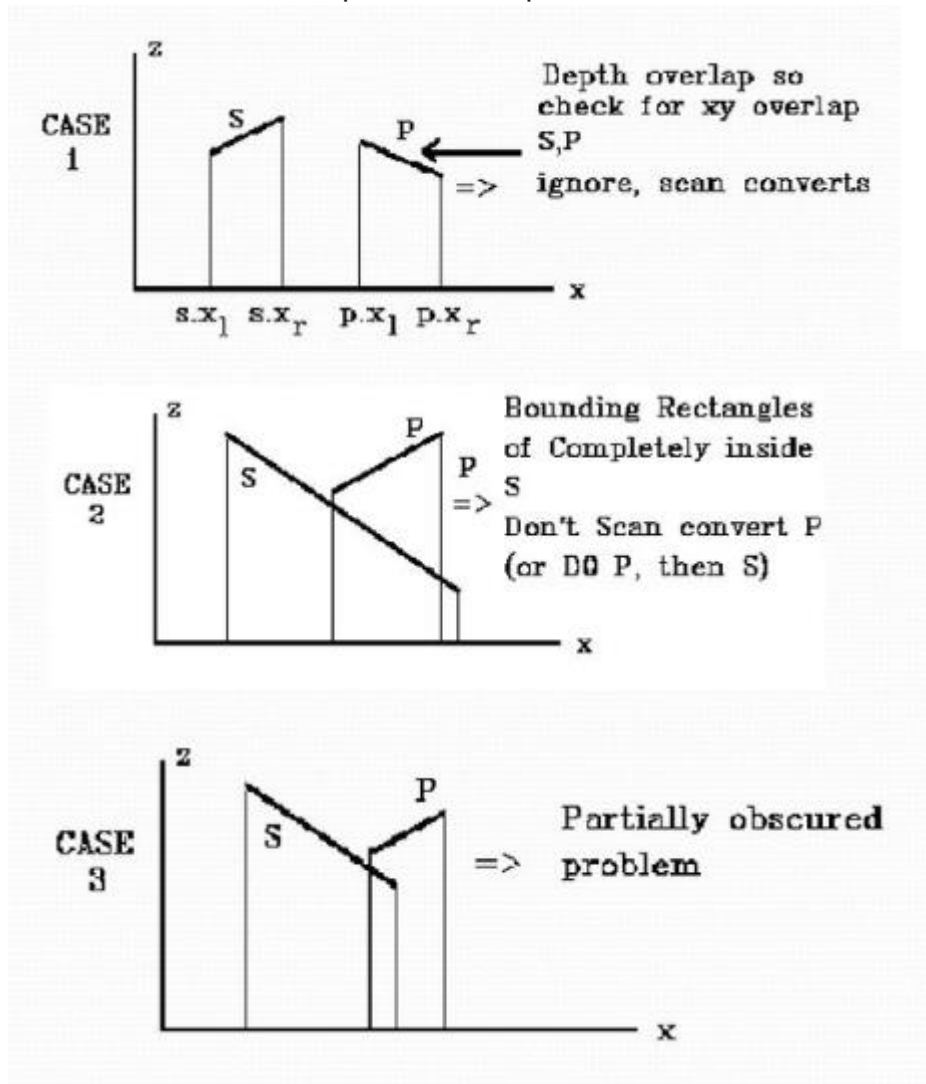
- Solución práctica, pero necesita procesamiento extra (tiempo).

Desventajas:

- El ordenamiento no es siempre posible.
- Ningún procesamiento extra sirve para casos cíclicos o de intersección.

Soluciones:

- Para cada polígono guardar su z mínima y máxima.
- Verificar que no haya traslapes entre polígonos con respecto a z .
- Si hay traslape en z se verifica que haya traslape en el plano xy . En este sentido existen 3 tipos de traslapado:



- En los casos 1 y 2 no hay problema.
- El caso 3 necesita procesamiento extra.

- Otra solución es dividir los polígonos traslapados en nuevos polígonos.

3. Algoritmo Z-Buffer: El algoritmo registra la coordenada z (profundidad) menor para todos los puntos que son visibles desde el punto (x, y) . Los valores se guardan en un buffer llamado buffer Z . Este algoritmo sigue los siguientes pasos:

- Inicializar el buffer z con el valor del plano lejano (o infinito).
- Para cada polígono P en la escena: Proyectarlo en la pantalla, luego calcular $Z(x, y)$ para cada pixel dentro del polígono: si $Z(x, y) < Z_{buf}(x, y)$ hacer $Z_{buf}(x, y) = Z(x, y)$ y pintar el pixel en (x, y) con el color de P en (x, y) .

Ventajas:

- Es fácil de implementar.
- Los requerimientos de memoria se pueden superar si la imagen se convierte por zonas.
- Puede utilizarse para cualquier tipo de objeto si se puede determinar el valor de z y la iluminación en cada punto de la proyección; observar que también es adecuado para objetos con superficies curvas ya que encuentra la superficie más cercana basándose en un test punto por punto.
- Los polígonos no tienen que compararse en un orden predeterminado: no es necesario un ordenamiento en z .
- No se requiere algoritmo de intersección.

Desventajas:

- No es eficiente. Puede pintar el mismo pixel varias veces.
- El buffer ocupa mucho espacio (una entrada por cada pixel de la pantalla).
- No maneja polígonos transparentes de forma adecuada.
- Tiene un problema de precisión debido al acortamiento en perspectiva.

Existen más métodos como el algoritmo Scan-Line, algoritmo de subdivisión, árboles BSP, entre muchos otros pero principalmente se encuentran estos 3 métodos muy utilizados para la eliminación u ocultamiento de caras no visibles para mejorar y optimizar el uso de recursos a la hora de renderizar una escena ya sea en 2D o 3D.

Fuentes:

<https://slideplayer.es/slide/16532347/>

<https://www.scribd.com/document/37538262/Ocultamiento-de-Superficies>

<https://cs.uns.edu.ar/cg/clasespdf/3.3-CO.pdf>