Métodos para detección de superficies visibles (Hearn-Baker)

Parte II

Contenido

- Método scan-lines
- Método ordenamiento en produndidad (del pintor)
- Método binary space partition tree (BSP-tree)
- Método basado en subdivisión de áreas
- Método basdo en octrees
- Método ray casting
- Métodos para superficies curvas
- Comparación

Método scan-lines

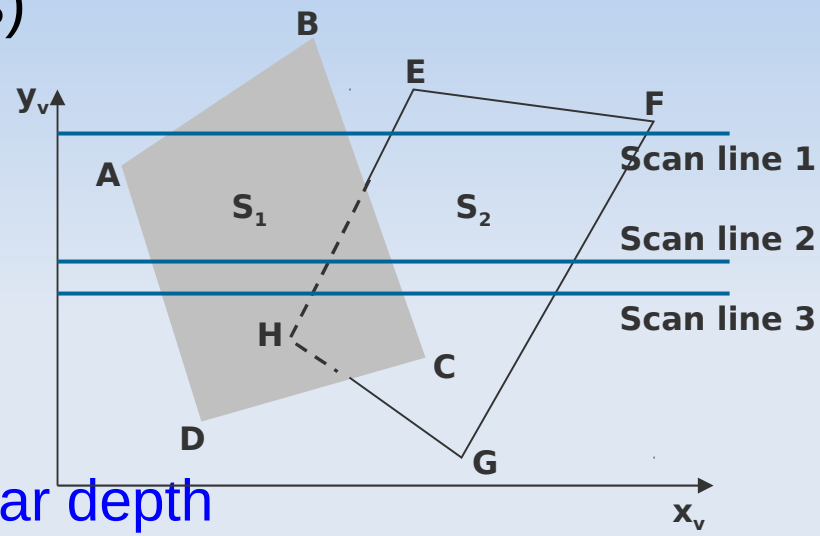
- Extensión del algoritmo para rellenar el interior de los polígonos
  - Por cada polígono que intersecta la scan-line
    - Procesar de izquierda a derecha
    - Calcular profundidad (depth) en caso que hay polígonos superpuestos
    - La intensidad del punto más cercano se almacena en el frame buffer

Método scan-lines: tablas

- Tabla de arcos
  - Puntos extremos de cada arco
  - Pendiente de la línea que pasa por el arco
  - Indices (punteros) a los polígonos que lo contienen
- Tabla de polígonos
  - Coeficientes de la ecuación del plano
  - Información de los colores (intensidad) del polígono
  - Indices (punteros) a la tabla de arcos

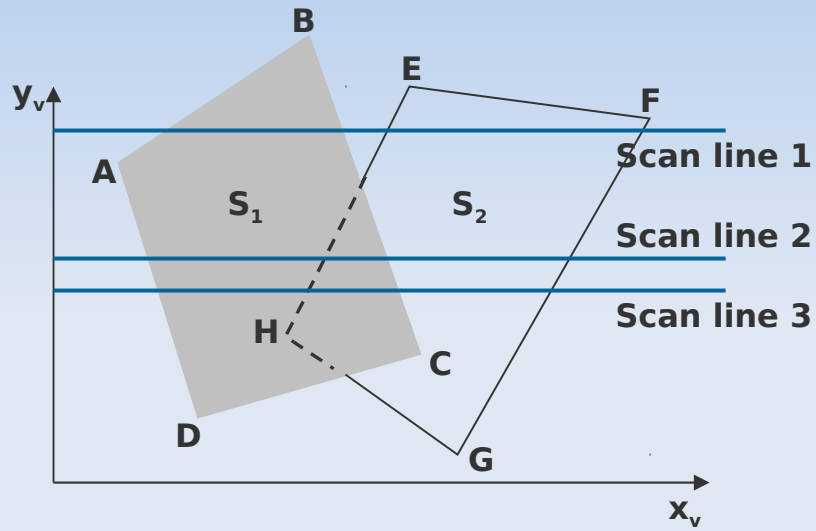
Método scan-lines: ejemplo

- Lista activa para scan-line 1 (arcos)
- Tabla de arcos:
  - AB, BC, EH, FG
  - Entre AB y BC solo el flag de S_1 está encendido
    - No es necesario calcular depth
    - El color de S_1 se utiliza
  - De manera similar entre EH y FG solo el flag de S_2 está encendido



Método scan-lines: ejemplo

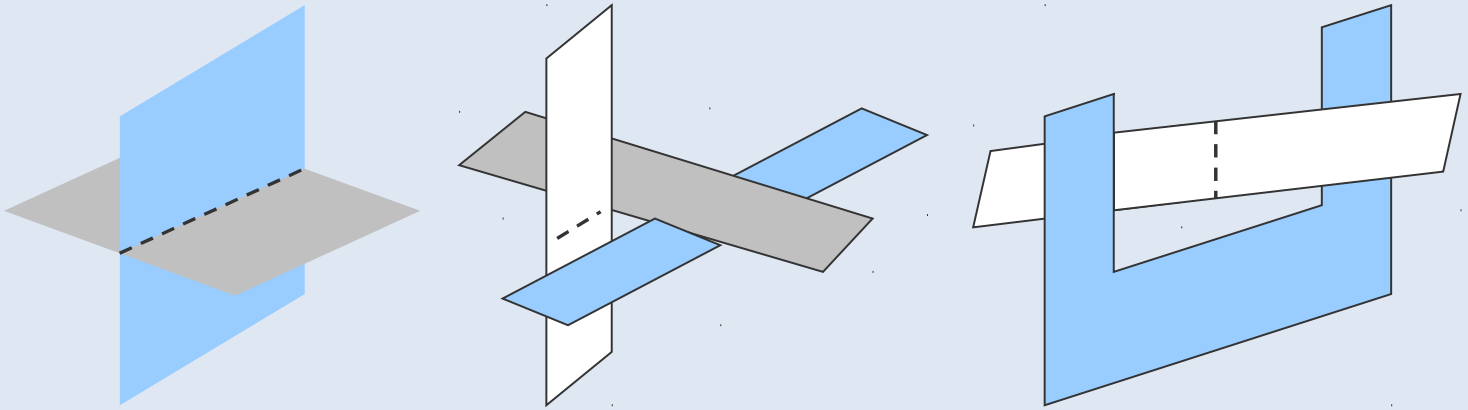
- Lista activa para scan-line 2 y 3
  - AD, BC, EH, FG
  - Entre AD y EH solo el flag de S_1 está encendido
  - Entre EH y BC ambos flags de S_1 y S_2 están encendidos



  - Calcular depth (profundidad de ambas superficies)
  - El color de S_1 se utiliza
- Obtener ventaja de la coherencia:
  - Pasando de una scan-line a la siguiente
  - Scan-line 3 tiene la misma lista activa que la 2
  - No es necesario hacer cálculos de profundidad entre EH y BC

¿Cuándo hay problemas?

- Este método funciona si polígonos no se superponen cíclicamente
  - Si de producen estos casos => dividir los polígonos



Método ordenamiento en profundidad

- Operaciones en el espacio de las imágenes y en el espacio de los objetos
  - Operaciones de ordenamiento en el espacio de las imágenes y en el espacio de los objetos
  - Scan-conversion en el espacio de las imágenes
- Funciones básicas:
  - Superficies (polígonos) ordenadas en forma decreciente
  - Scan-conversion sobre las más lejanas primero (valor de depth más grande)

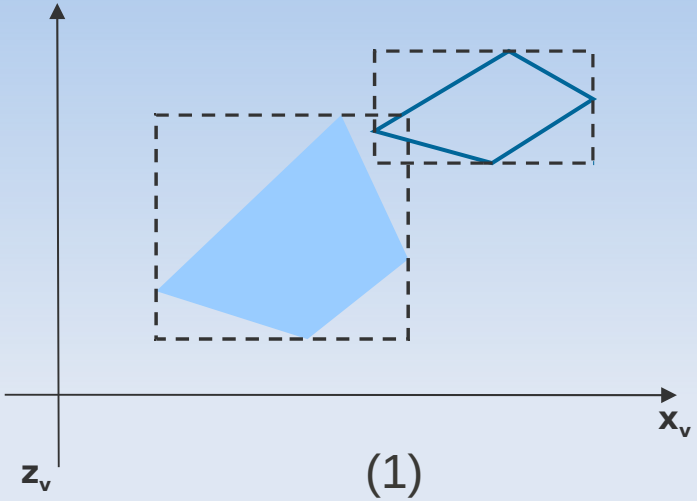
Algoritmo

- Conocido como algoritmo del pintor
- Para crear un cuadro:
  - Primero pinta los colores del background
  - Luego los objetos más distantes
  - A continuación los que los siguen en cercanía y asi sucesivamente
  - Finalmente se pinta la superficie mas cercana (foreground)
- Proceso:
  - Ordenar los polígonos (superficies) de acuerdo a su distancia el viewplane
  - Las intensidades(colores) de la más lejana se ingresa el frame buffer
  - Tomar la superficie siguiente y repite el proceso

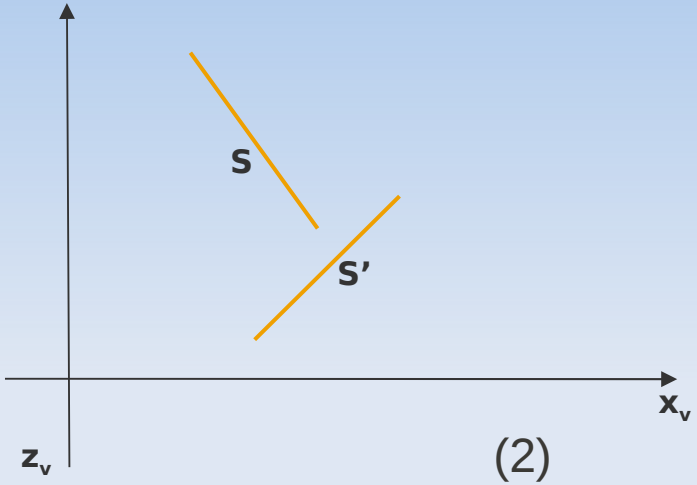
Algoritmo

- Este proceso se realiza en varios pasos:
  - Superficies ordenadas con respecto al z-value mas pequeño
  - La superficie S con el z-value más grande es analizada para ver si se superpone con otras
    - Si no hay superposición en el eje z => scan-conversion
    - Si hay superposición en el eje z => analizar si las superficies deben ser reordenadas

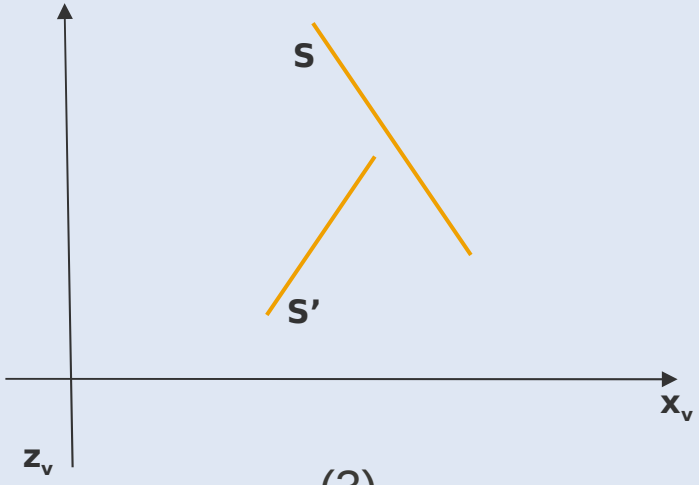
Algoritmo: ejemplos de overlapping usando una vista solo de los ejes x y z



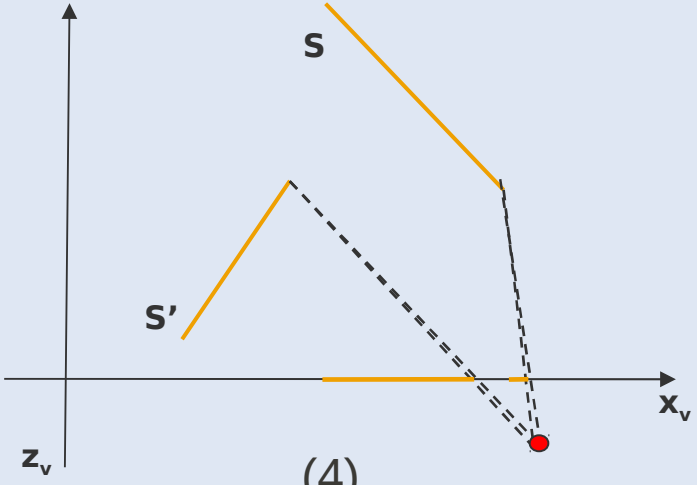
(1)



(2)



(3)



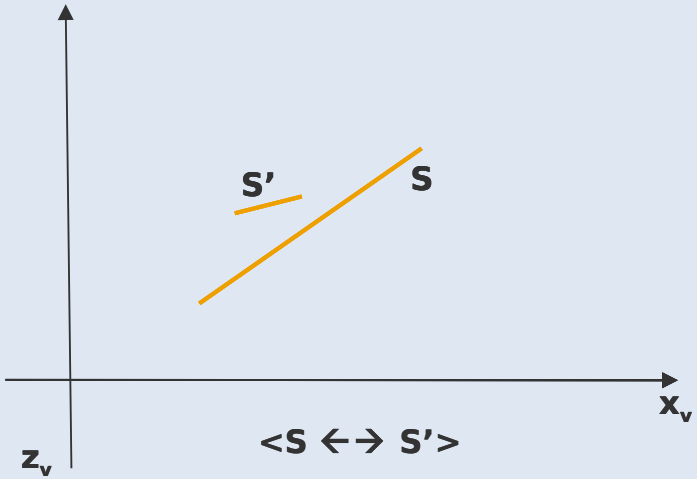
(4)

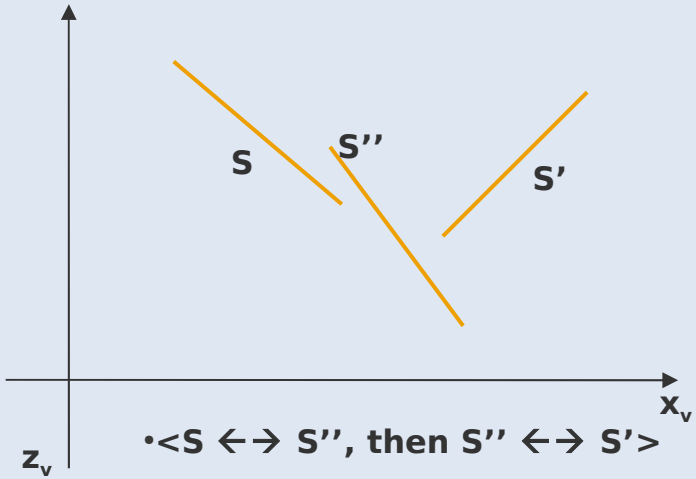
Algoritmo: ejemplos de overlapping

- Si S pasa alguno de los siguientes tests (son verdaderos), no es necesario reordenar
  - Los bounding rectangles (coordinate extents) de las dos superficies no se superponen en el plano XY (Figura 1)
  - La superficie S está completamente detrás de la otra mirada desde la posición de viewing (Figura 2)
  - La superficie que se superpone está completamente adelante de S desde la posición de viewing (Figura 3)
  - La proyección de los arcos del borde de cada polígono no se superponen
- Si los 4 tests fallan? Qué casos existen?

Algoritmo: ejemplos de overlapping

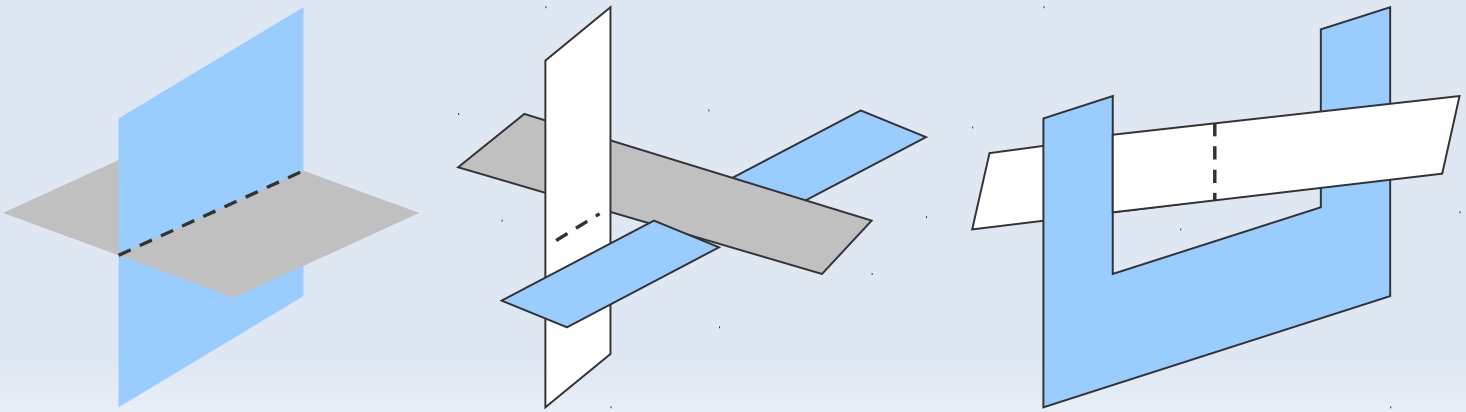
- Si los 4 test fallan con S':
  - Intercambiar S y S' en la lista ordenada
  - Repetir los tests para cada superficie que es re-ordenada en la lista





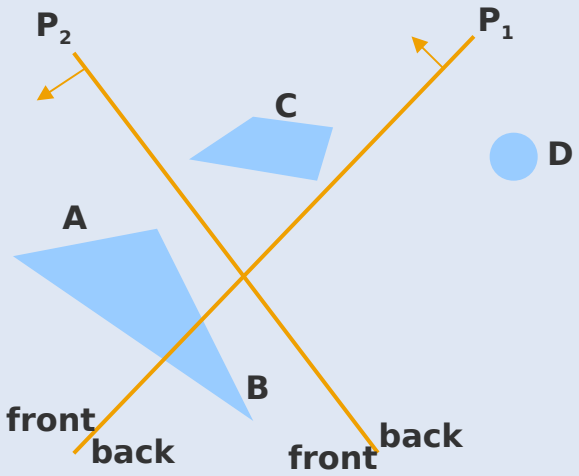
Algoritmo: problemas

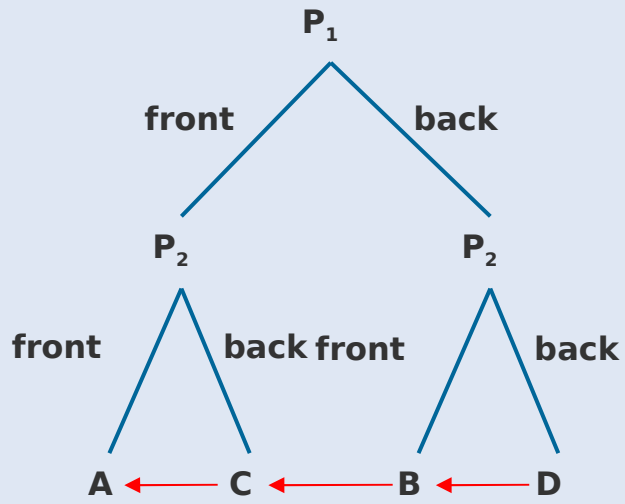
- Si dos o mas caras se “tapan” una con otra
  - Loop infinito
  - Como evitarlo: marcar cada cara cuando ha sido reordenada a una profundidad más lejana
    - No puede ser reordenada de nuevo
  - Si intenta reordenarla de nuevo:
    - Dividir el polígono en dos polígonos más pequeños
    - La superficie original es ahora reemplazada por dos nuevas superficies



Método de partición binaria del espacio (BSP)

- Determina la visibilidad de los objetos pintando las caras desde la más profunda hasta la más cercana (parecido al algoritmo del pintor)
  - Identificar superficies(polígonos) fuera o dentro del plano de partición
  - Dividir los objetos intersectados por el plano si es necesario



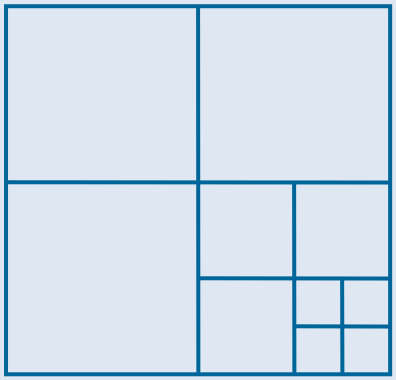


Método de subdivisión de áreas

- Saca ventaja de las coherencia de áreas
  - Creando áreas visibles que representan una parte de una única superficie
  - Dividiendo sucesivamente el área de viewing en rectángulos pequeños
    - Hasta que cada área es la proyección de una sola superficie o de ninguna
  - Qué problemas que hay resolver (tests de identificación)?
    - Identificar un área con una superficie
    - Identificar si el área es demasiado compleja de analizar
- A qué estructura se parece?

Método de subdivisión de áreas:algoritmo

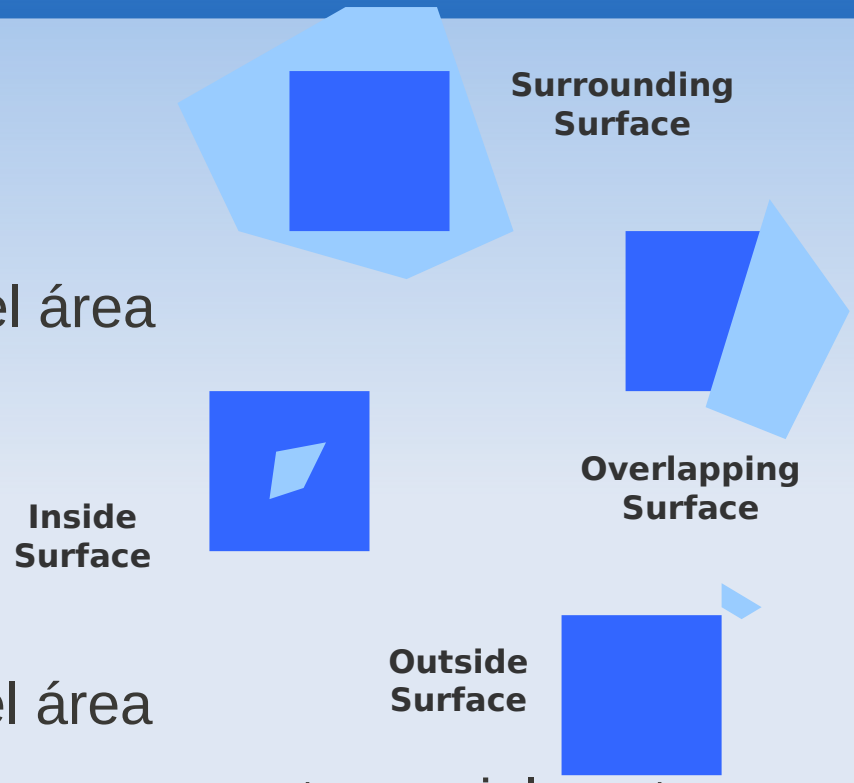
- Similar a construir un quadtree
- Comenzando con la vista completa
  - Aplicar los tests de identificación
  - Si los tests indican que la vista dentro de un área es lo suficientemente compleja
    - Subdividir
  - Aplicar ls tests a cada área más pequeña



    - Hasta que contenga una sola superficie
    - Hasta que su tamaño sea de un único pixel
    - Ejemplo: con resolución 1024x1024, si se divide 10 veces, el tamaño de cada área es de un pixel

Método de subdivisión de áreas: tests de identificación

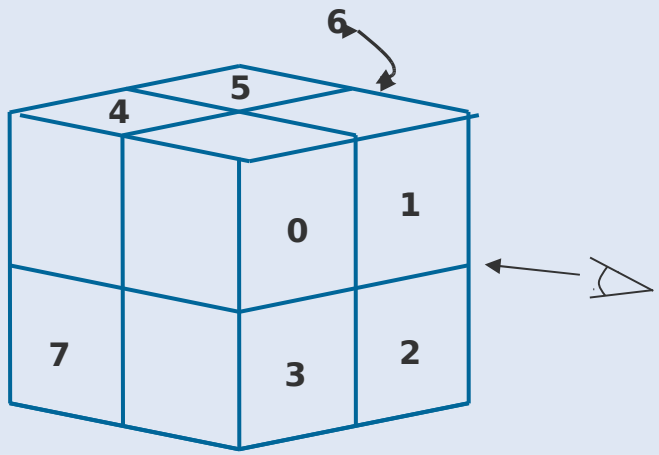
- Cuatro posibles relaciones
  - Superficie rodea área
  - Superficie se superpone en parte del área
  - Superficie dentro del área
  - Superficie fuera del área



- Un área no se divide si:
  - Todas las superficies están fuera del área
  - Existe solo una superficie rodeando, superpuesta parcialmento o dentro del área
  - Una superficie que rodea el área oculta todas las demás que la intersectan

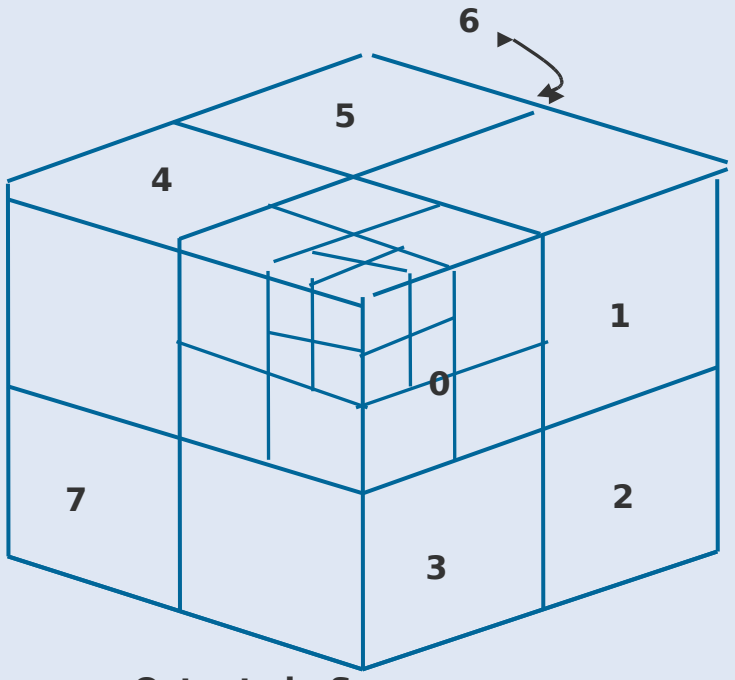
Método de subdivisión basado en octrees

- Extensión del método de división de áreas
- Se proyectan los nodos en el view plane
  - Orden desde adelante hacia atrás (se recorren encuentran y analizan primero las superficies menos profundas
  - Los nodos de los primeros octantes son visitados primero que los nodos posteriores
    - A un pixel se le asigna el color de la superficie que se encuentra en el primer nodo que le corresponde



Método de subdivisión basado en octrees

- Cómo se dibuja?
  - Asociar (map) el octree a un quadtree de área visible
    - Recorrer los nodos del octree, recursivamente, desde adelante hacia atrás
    - La información del quadtree es cargada en el frame-buffer



Método ray casting

- Basado en métodos ópticos geométricos
- Trazar caminos de rayos de luz
  - Trazar un rayo desde cada pixel, en la línea de visión, para intersectar la escena
  - Determinar qué objetos intersectan esta escena
  - Identificar la superficie visible cuyo con puntos de intersección más cercano al plano de visión (view plane)
- Considerar solo los rayos que pasan por un pixel
- Nota: Método efectivo para escenas con superficies curvas

Clasificación de los métodos

- Métodos en el espacio de las imágenes
  - Depth-buffer
  - A-Buffer
  - Scan-line
  - Subdivisión de áreas
- Métodos en el espacio de los objetos
  - Detección de Back-faces
  - BSP-tree
  - Subdivisión de áreas
  - Octrees
  - Ray casting

Métodos para superficies curvas

- Métodos efectivos para superficies curvas
  - Ray casting
  - Octrees
- Aproximar las superficies curvas con un conjunto de caras poligonales planas
  - Aplicar alguno de los métodos vistos
- Más eficiente y preciso es usar ray casting y las ecuaciones que describen las superficies

Comparación

- Método back-face
  - Bueno y efectivo para eliminar rapidamente las caras que no se ven pero no resuelve el problema completamente
- Z-buffer
  - Rápido y simple
  - Dos buffers (colores y profundidad)
- A-buffer
  - Mejora el z-buffer incluyendo transparencia, antialiasing,...
- Con respecto a la complejidad de la escena medido en la cantidad de polígonos que se superponen :
  - Escenas simples (hasta mil polígonos): ordenamiento en profundidad, BSP tree, scan-lines, z-buffer
  - Escenas complejas (más de mil): z-buffer, octrees