

UNIDAD 2

2 Definiciones básicas del modelado de sólidos

2.1 CSG

2.1.1 Definición de CSG

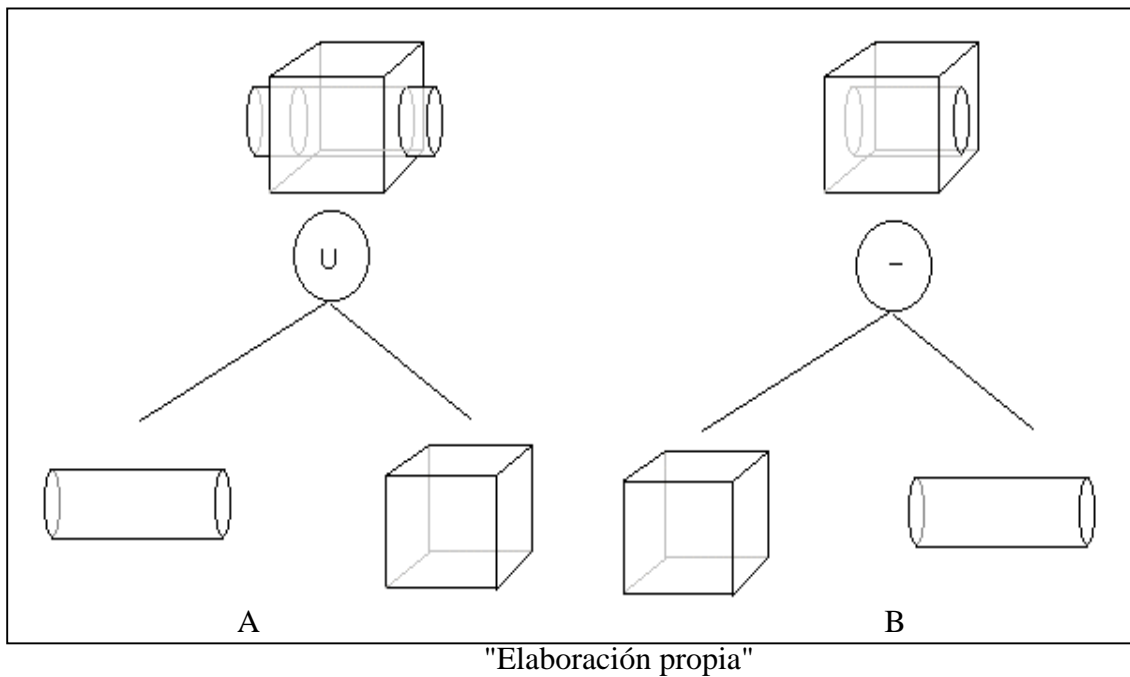
CSG viene de las siglas en Inglés Constructive Solid Geometry (Geometría Sólida Constructiva). CSG se define como la representación de un objeto mediante un árbol binario, donde los nodos no terminales representan operaciones Booleanas (unión, diferencia, intersección y complemento) y los nodos hojas representan sólidos primitivos [6]. Por lo tanto, cuando se representa un objeto mediante árboles CSG se utilizan una serie de primitivas que al combinarse por operadores Booleanos generan el objeto deseado.

Las primitivas sólidas de CSG son sólidos simples tales como paralelepípedos, cilindros, semi-espacios, y esferas mediante los cuales se crean operaciones Booleanas las cuales se pueden calcular mediante ecuaciones simples, por ejemplo el semi-espacio está dado por la fórmula $Ax+By+Cz+D \leq 0$.

En la figura 2.1 se muestran dos objetos definidos por medio de la representación CSG. El primero está formado por dos primitivas: un cilindro y un paralelepípedo, los cuales se unen para formar un solo objeto. El segundo árbol está formado por las mismas primitivas pero ahora primero la primitiva paralelepípedo y después la primitiva cilindro y

utilizando el operador Booleano diferencia. Como se puede apreciar en la figura, los dos objetos son diferentes, el primer objeto es un paralelepípedo con un cilindro atravesado mientras que el segundo objeto, es un paralelepípedo con un hueco adentro de él, en forma de cilindro.

Figura 2.1: A)Árbol de un objeto definido vía CSG de la unión de dos primitivas.
 B)Árbol de un objeto definido vía CSG de la diferencia de dos primitivas



Entonces se puede apreciar que un árbol CSG aún con las mismas primitivas puede formar diferentes objetos sólo con cambiar los operadores booleanos y/o el orden de las primitivas.

El esquema CSG se expresa recursivamente en la forma de Backus Naur [15] como:

$$\begin{aligned} \text{Árbol CSG} ::= & \quad \langle \text{Árbol CSG} \rangle \langle \text{Op Bool} \rangle \langle \text{Árbol CSG} \rangle \\ & \quad | \langle \text{Árbol CSG} \rangle \langle \text{Transformación} \rangle \end{aligned}$$

| <Sólido Primitivo>

| <Semi-espacio>

Op Bool ::= Unión | Intersección | Diferencia

Transformación ::= Translación | Rotación | Escalamiento | ...

Sólido Primitivo ::= Paralelepípedo | Cilindro | Esfera | Cono | ...

Debido a que mediante el esquema de Backus Naur, se crea el árbol Booleano, este método se puede utilizar para almacenar un objeto, y cuando se requiera se puede crear el Árbol CSG del objeto. Aunque los árboles n-arios algunas veces se usan por compatibilidad con entradas de usuarios más naturales son equivalentes, y se pueden convertir o procesar directamente como árboles binarios [7,8]

2.1.2 Características formales de CSG

Los árboles CSG son concisos, no ambiguos y cerrados, pero no son únicos. Estos son fáciles de crear y editar. Su dominio depende de conjunto de objetos primitivos disponibles, así como del conjunto de operadores Booleanos y de movimiento. Solo se necesita llevar a cabo una inspección sintáctica local simple para validar un árbol CSG (el cual siempre está acotado, si sus primitivas están acotadas) [1].

2.2 Representación de división espacial

En las representaciones de división espacial, un sólido se descompone en una colección de sólidos adyacentes, que no se intersectan y, que son más primitivos que el sólido original, y no necesariamente del mismo tipo. Las primitivas pueden variar en forma y parámetros (tamaño, posición y orientación). Las representaciones de división espacial más comunes son [2]:

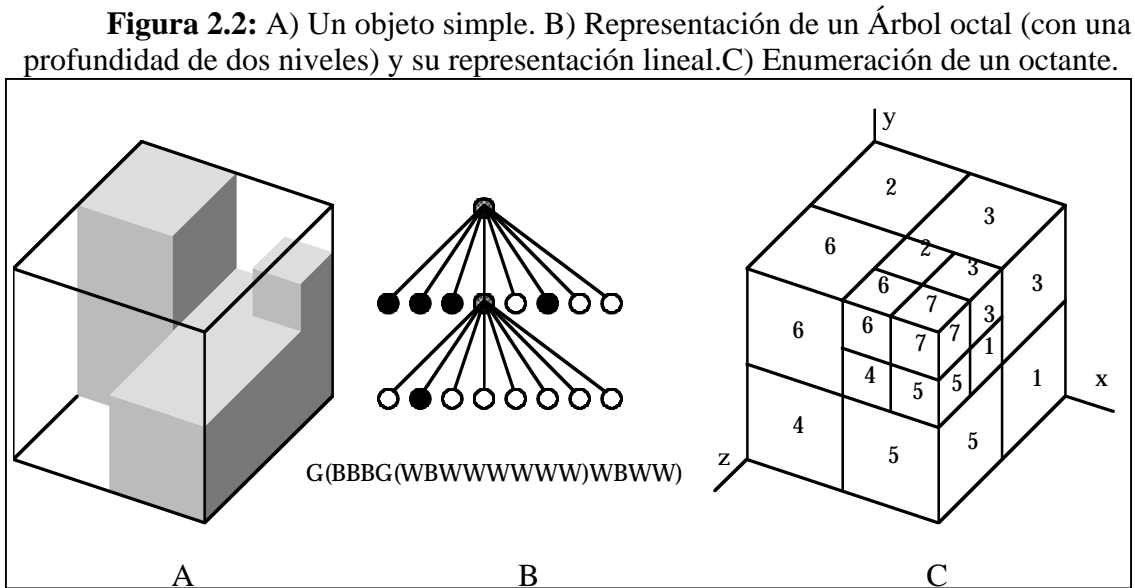
- Árboles octales clásicos.
- Descomposición en celdas.
- Enumeración espacial o voxelización.

La representación Árboles Octales Clásicos será analizada más cuidadosamente, debido a que es la representación de interés, las otras dos serán descritas brevemente sólo para conocimiento.

2.2.1 Árboles octales clásicos (Classical Octrees).

Un árbol octal sirve para representar volúmenes y este tipo de árbol tiene la característica de que cada nodo excepto las hojas tiene 8 hijos. Cuando se trata de representar un objeto sólido, a los nodos se les conoce como octantes, y un nodo interno debe tener al menos un octante hijo diferente a los demás, de lo contrario este nodo interno tendrá que ser un nodo hoja. También es importante notar que un nodo hoja sólo puede tener dos posibles valores: negro o blanco, mientras que el nodo padre debe ser gris. Hay

dos formas de representar un árbol octal, por medio de apuntadores o por medio de una representación lineal [9]. En la Figura 2.2 se representa en A) un ejemplo de un objeto simple, en B) el árbol octal del objeto por medio de apuntadores y por medio de la representación lineal; en C) se muestra la enumeración de un octante.



El dibujo se obtuvo de [1] Pág. 22

Existe una convención llamada ZYX, en la cual se determina que la enumeración de los nodos se hace del 0 al 7. En la figura 2.2 C se muestra la enumeración de los nodos. Y se representa con 3 bits (000, 001, 010, 011, 100, 101, 110 y 111), donde el primer bit representa la coordenada Z, el segundo bit representa la coordenada Y y el último bit es la representación de la coordenada X [16]. Un nodo (negro, blanco o gris) se puede almacenar en dos bits [10] por ejemplo 01 para negro, 00 para blanco y 10 u 11 para gris.

Un Árbol Octal tiene diferentes aplicaciones, como por ejemplo para representar la relación espacial de objetos geométricos, o para representar volúmenes, cuando se utiliza

para representar objetos sólidos, el árbol se va a crear de acuerdo a la relación de cada octante y el volumen deseado. Para crear un árbol octal para construir objetos sólidos se utiliza un procedimiento recursivo, el cual está descrito en [5]: "Se inicia desde un cubo el cual representa el espacio del modelo (profundidad 0), si el objeto contenido en este cubo es muy complejo, el cubo será clasificado como gris y será dividido en ocho octantes, entonces se estudiarán cada uno de estos octantes en un orden preestablecido (profundidad 1). En la codificación del Árbol Octal Clásico, el proceso de subdivisión se repite hasta que se obtengan octantes blancos y negros, o cuando una mínima escala es alcanzada (profundidad = profundidad máxima). Un octante negro se define como un octante que está completamente dentro del objeto, mientras que un octante blanco está completamente fuera".

La propiedad esencial de los Árboles Octales es que almacenan la información de la forma del objeto en una manera ordenada espacialmente, y la naturaleza altamente jerárquica de éstos sugiere el uso de paradigmas recursivos "Divide y vencerás" [1]

Esta propiedad es muy útil cuando se desea visualizar un objeto, ya que el árbol octal únicamente se recorrerá en cualquiera de las formas mostradas en la Tabla 2.1, dependiendo de las caras que se deseen visualizar. En la tabla en la columna "Caras visibles", B (Back) es la cara de atrás de un octante, F (Front) es la cara de enfrente del octante D (Down) es la cara de abajo, U (Up) es la cara de arriba, L (Left) es la cara izquierda y R (Right) es la cara derecha. Entonces si por ejemplo se desea visualizar las caras de atrás, abajo e izquierda (B, D, L) de un objeto únicamente se debe seguir el recorrido necesario para visualizar esas caras en este caso 7, 6, 5, 3, 4, 2, 1 y 0.

Tabla 2.1 Enumeración de atrás para adelante y caras visibles

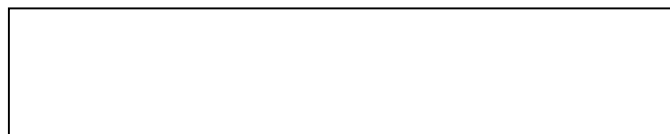
Orden de atrás para adelante	Caras visibles
7, 6, 5, 3, 4, 2, 1, 0	B, D, L
6, 7, 4, 2, 5, 3, 0, 1	B, D, R
5, 4, 7, 1, 6, 0, 3, 2	B, U, L
4, 5, 6, 0, 7, 1, 2, 3	B, U, R
3, 2, 1, 7, 0, 6, 5, 4	F, D, L
2, 3, 0, 6, 1, 7, 4, 5	F, D, R
1, 0, 3, 5, 2, 4, 7, 6	F, U, L
0, 1, 2, 4, 3, 5, 6, 7	F, U, R

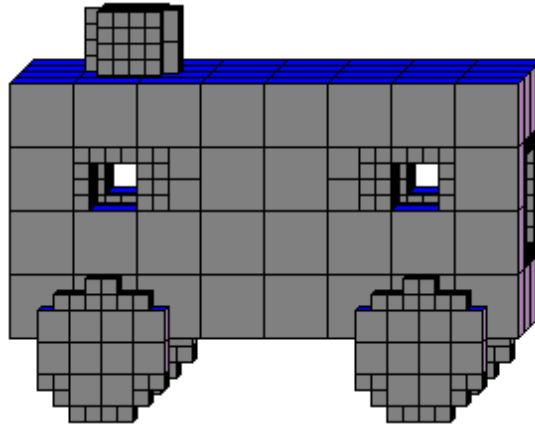
Esta tabla se obtuvo [2]. Pág. 696.

La Figura 2.3 muestra un ejemplo de la visualización atrás para adelante, la forma en que se visitan los nodos es 0, 1, 2, 4,3, 5, 6 y 7, y como se puede observar de la figura se visualizan únicamente las caras de arriba, derecha y de enfrente de cada octante, y de acuerdo a como se va haciendo la visualización los octantes de atrás de la figura se van tapando con los octantes más cercanos.

El conjunto de operaciones Booleanas se puede utilizar directamente en los árboles octales[14], únicamente es necesario hacer el recorrido paralelo entre dos árboles octales para así poder crear un nuevo árbol octal con las características dadas por los dos árboles, esto se aplica para la unión, la intersección y la diferencia; para el complemento únicamente se van a cambiar los valores de los octantes hojas por su complemento, es decir para un nodo hoja negro se genera un octante blanco y para un nodo hoja blanco se genera un octante negro.

Figura 2.3: Visualización de un objeto codificado por Árbol Octal usando un algoritmo de Visualización atrás para adelante. Cada uno de los octantes del Árbol Octal fue dibujado con sus bordes, de esta forma el tamaño de cada nodo se puede visualizar.





"Elaboración propia"

En las tablas 2.2, 2.3, 2.4 y 2.5 se muestran los resultados de las operaciones Booleanas dependiendo el valor de cada nodo.

Tabla 2.2: Complemento de un nodo.

Nodo	Complemento
N	B
B	N
G	G*

La tabla se obtuvo de [1] Pág. 24

Tabla 2.3: Unión entre dos nodos.

U	B	W	G
B	B	B	B
W	B	W	G
G	B	G	G*

La tabla se obtuvo de [1] Pág. 24

Tabla 2.4: Intersección entre dos nodos.

I	B	W	G
B	B	W	G
W	W	W	W

G	G	W	G*
---	---	---	----

La tabla se obtuvo de [1] Pág. 24

Tabla 2.5: Diferencia entre dos nodos.

D	B	W	G
B	W	B	G
W	W	W	W
G	W	G	G*

La tabla se obtuvo de [1] Pág. 24

Donde:

B = Nodo negro

W = Nodo blanco

G = Nodo gris

G* = caso recursivo del algoritmo

U = Unión

I = Intersección

D = Diferencia

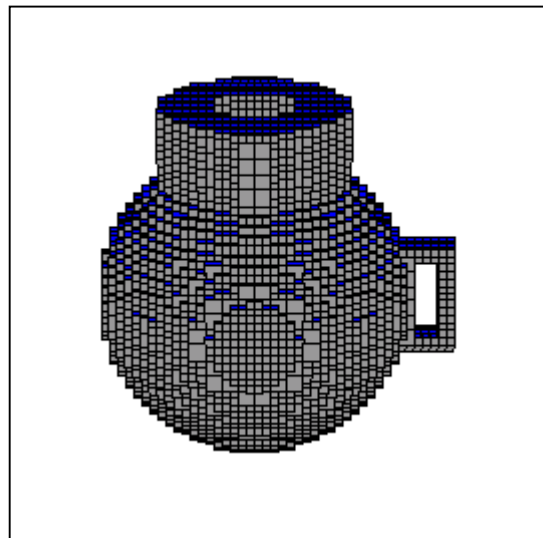
Los resultados de las operaciones Booleanas van a depender de cómo sean los dos árboles octales que se ocupan como operando. Por ejemplo se tiene un octante negro y un octante gris y se va a realizar una unión, entonces no es necesario realizar el sub-árbol octal, y hacer las operaciones sobre este, si no que el resultado se reduce al octante menor. Aunque no siempre ocurre, por ejemplo si se tiene la unión entre un sub-árbol y un octante blanco, es necesario generar todo el sub-árbol en el nuevo árbol.

No hay garantía de que al realizar estas operaciones entre dos árboles octales, el árbol resultante quede perfectamente creado, es decir: Tener nodos grises internos cuyos

nodos hijos son todos negros o todos blancos [1]. Estos nodos, sin embargo, se pueden cambiar con un algoritmo transversal del árbol, relativamente fácil [5].

Entre más precisión se requiera cuando se construye el árbol octal de un objeto mayor es el número de octantes que se van a utilizar, y los octantes que logran esta precisión, son los octantes más pequeños, los cuales se encuentran en los bordes de los objetos como se muestra en la Figura 2.4. Estos octantes ocasionan que aumente el uso de memoria y la mayoría de veces hay más octantes en el borde de una figura que en el resto del objeto.

Figura 2.4 Objeto con los bordes de cada octante



"Elaboración propia"

2.2.2 Descomposición en celdas.

Una de las formas más generales de división espacial es la Descomposición en celdas. Cada sistema de descomposición en celdas define un conjunto de celdas primitivas que están típicamente parametrizadas en formas específicas [2].

Mediante esta representación se puede construir objetos más complejos desde objetos más simples, por ejemplo al juntarlos.

La representación de un objeto en la descomposición de celdas es no ambigua, y no necesariamente es única, como se muestra en la figura 2.5. Las representaciones por descomposición en celdas son difíciles de validar, pues cada par de celdas deben ser probadas para intersecciones potenciales [2].

Figura 2.5: Las celdas mostradas en (A) pueden ser transformadas para construir el mismo objeto mostrado en (B) y en (C) de formas diferentes. Incluso un solo tipo de celda es suficiente para causar ambigüedad.

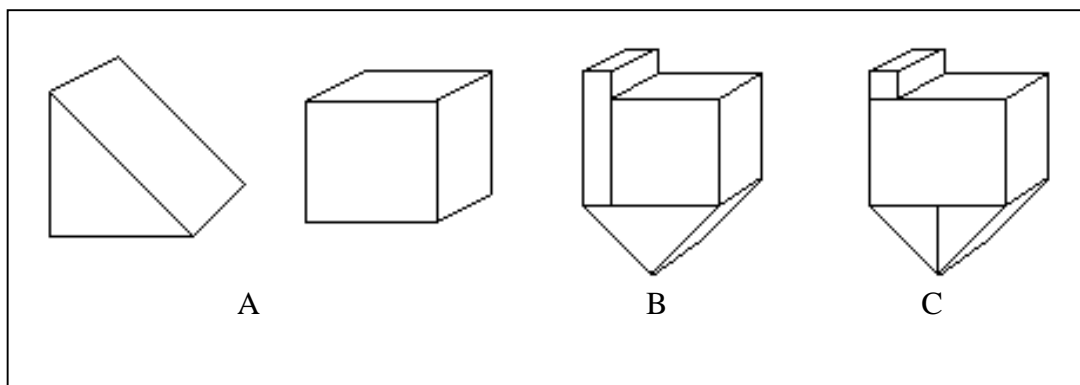


Figura obtenida de [2] Pág. 549

2.2.3 Enumeración espacial o voxelización.

La enumeración espacial (también se conoce como enumeración exhaustiva) es un caso especial de la descomposición de celdas en el que el sólido se descompone en celdas idénticas, acomodadas en una cuadrícula regular y fija [6]

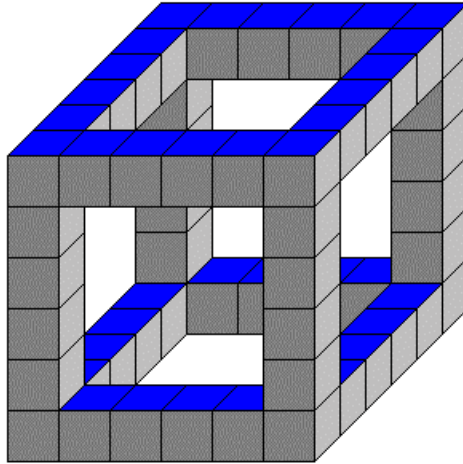
El tipo de celda más común es el cubo, y la representación del espacio como un arreglo de cubos se llama voxelización [2].

Este método es simple porque únicamente se va determinar si una celda esta siendo ocupada o no, en la cuadrícula. Entonces cuando se representa un objeto solo se debe decidir qué celda se encuentra ocupada y marcarse como llena, o si no se ocupa debe marcarse como vacía. Se puede ver claramente entonces, que para obtener una precisión deseada únicamente va a cambiar el tamaño de la cuadrícula, de esta forma entre más pequeña sea la cuadrícula más precisión tendrá un objeto realizado por este método, pero también la memoria tiende a aumentar. Debido a estas características se puede almacenar un objeto por medio de un arreglo de tres dimensiones y por lo tanto no es tedioso obtener qué voxels forman a un objeto. Una desventaja es que, como no existe el concepto de celda parcial, los objetos únicamente se van a aproximar, o en el peor de los casos se tendrán celdas muy pequeñas y un aumento considerable de memoria.

La figura 2.6 muestra un objeto que está representado por enumeración espacial. Las celdas presentes son las que se visualizan.

Figura 2.6: Un objeto sólido que está representado por enumeración espacial.





"Elaboración propia"

2.3 Conclusiones

Para crear gráficas por medio de computadora, existen gran cantidad de métodos, aquí solo se nombraron algunos. Todos los métodos tienen ventajas y desventajas y deberán de elegirse de acuerdo a las necesidades que se tengan, por ejemplo tal vez se necesite crear objetos con poco espacio, sin importar cuan precisos sean, o tal vez se necesitan crear objetos precisos sin importar la cantidad de tiempo que se lleve al crearlos.

El método Árbol octal es un buen método, por que al construir un objeto se puede subdividir únicamente la parte donde no se tenga la precisión adecuada, mientras que por ejemplo en la enumeración espacial es necesario dividir toda la cuadrícula, aunque en el método de enumeración espacial un objeto se puede obtener más fácilmente desde un arreglo tridimensional, sin necesidad de tener que hacer un análisis complejo.

El método Árbol octal al combinarse con la Geometría Sólida Constructiva es muy útil para crear objetos complejos, únicamente con objetos sencillos, y ofrece buena aproximación; aunque entre más complejo sea el objeto más tiempo se tardará en crear.