

Modelado Geométrico (2)

1 de Octubre, 2010

Informática Gráfica
Ingeniería Superior Informática



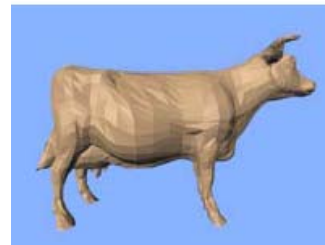
Índice

1. Introducción:
 - 1.1 Requisitos de las técnicas de representación
 - 1.2 Adquisición de objetos
2. Técnicas de representación de objetos
 1. Modelos de alambre
 2. Generación mediante primitivas parametrizables
 3. Representaciones de barrido y revolución
 4. Representaciones por fronteras. Modelos de mallas
 5. Representaciones de partición espacial
 6. Geometría constructiva de sólidos
 7. Modelos basados en funciones implícitas
3. Comparación entre representaciones

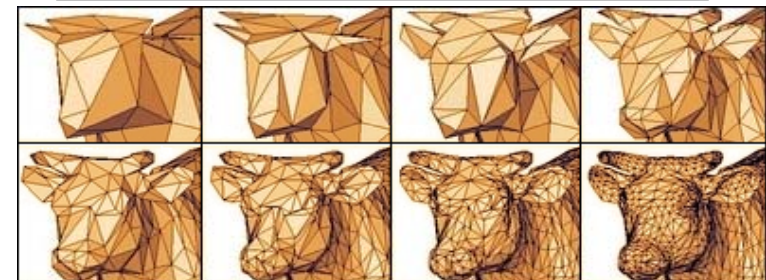
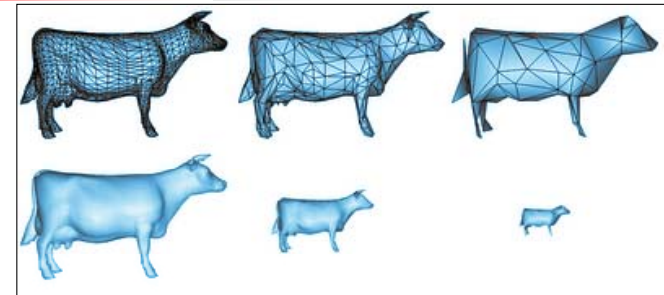


2.4 Representación de fronteras (B-Rep, boundary representation)

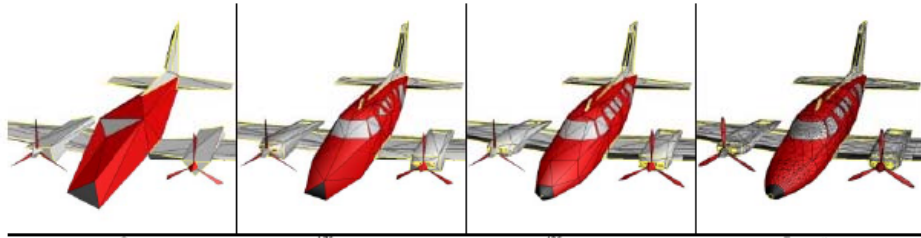
- Describen un objeto en función de sus fronteras superficiales, es decir de las caras y superficies que lo configuran.
- Pueden basarse en caras planas o curvas
- Restricciones adicionales:
 - Polígonos convexos
 - Triángulos(tiras/abanicos de triángulos: mayor eficiencia)
- Definición en función de sus vértices, aristas y caras (recordamos que pueden ser rectas o curvas)



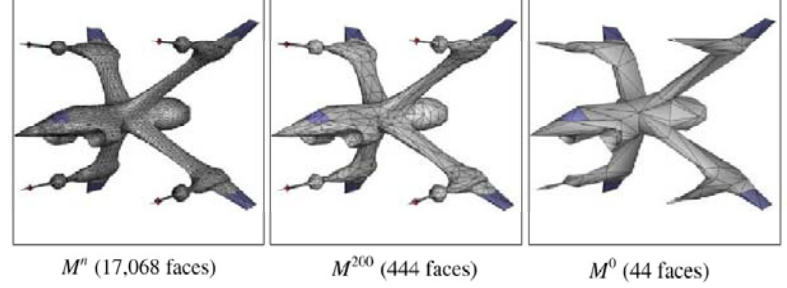
Mallas de polígonos



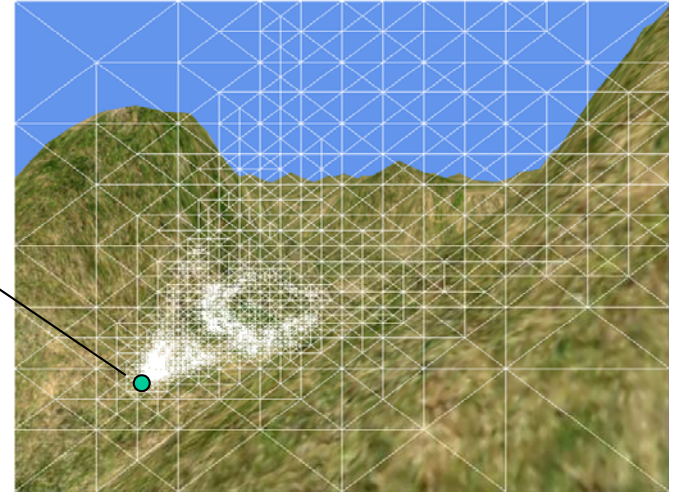
Modelos multirresolución



(a) Base mesh M^0 (150 faces) (b) Mesh M^{25} (500 faces) (c) Mesh M^{225} (1,000 faces) (d) Original $M=M^0$ (13,546 faces)
 Figure 5: The PM representation of an arbitrary mesh M captures a continuous-resolution family of approximating meshes $M^0 \dots M^p = M$.



Mallas progresivas

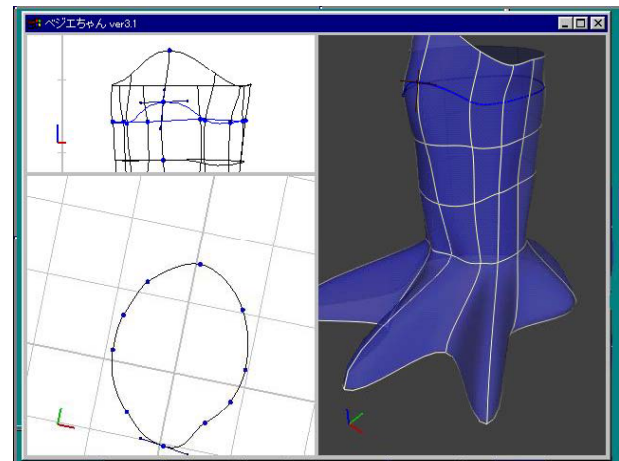


cámara

Sobre todo para terrenos, también para envío de mallas por internet...



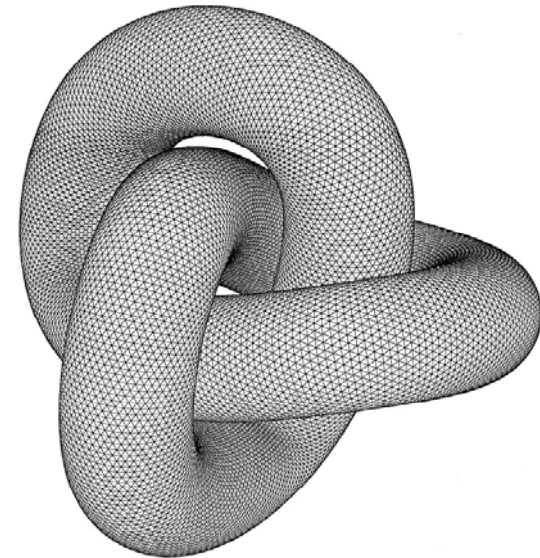
Parches bicúbicos



Parches con 4 vértices esquina. En la unión entre parches coinciden la posición y la derivada → la normal es continua, lo que hace que la superficie parezca mucho más suave.



Aproximación de superficies mediante mallas



Podemos utilizar mallas como "representación de máquina" en gráficos



Aproximación de superficies mediante mallas



Los algoritmos de sombreado recuperan la apariencia curva de las superficies



2.5 Representación de partición espacial

- Las representaciones de partición espacial forman una familia de métodos que se caracterizan por dividir el espacio en un conjunto de celdas, y estudiar cuáles están ocupadas y cuáles vacías.
- Cuando se quiere dibujar un objeto se recorren las celdas y se comprueba si ese objeto ocupa la celda o no, y en caso de ocuparla, si lo hace total o parcialmente.
- Más que una representación, las particiones espaciales son estructuras de datos que nos ayudan a operar con los modelos. Permiten encontrar primitivas del modelo en tiempo $O(\lg n)$ en lugar de $O(n)$.



Representación de partición espacial

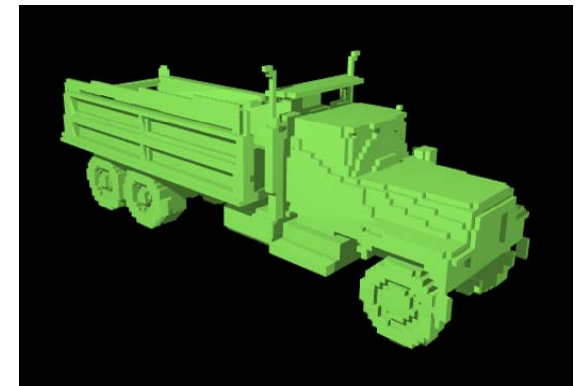
- Los métodos de ocupación espacial se diferencian en dos puntos fundamentales:
 - ¿Cómo dividir el espacio ?
 - ¿Qué hacer cuando se detecta una celda parcialmente ocupada ?
- Al dividir el espacio en celdas cúbicas los únicos objetos que se pueden representar con precisión son aquellos con superficies poliédricas.
- La única forma de aumentar la exactitud del modelo es incrementando la resolución, es decir dividir el espacio en celdas más pequeñas, aumentando el número de elementos de volumen o *voxels* necesarios para describir la escena.



Representación de partición espacial: Voxels

Se basan en descomponer la escena en un número prefijado de celdas idénticas, dispuestas sobre una malla regular fija.

El tipo de celda más común es el cubo (voxel)



Representación de partición espacial: Voxels

- Ventajas de los voxels:
 - Para representarlo sólo es necesario comprobar para cada celda si está ocupada o no.
 - En caso de estar ocupada se dibuja con el color del objeto, y si no, se deja vacía.
 - Un objeto se puede codificar con una lista única y no ambigua de celdas ocupadas.
 - Es sencillo determinar si una celda está dentro o fuera del sólido, y también si dos objetos son adyacentes.



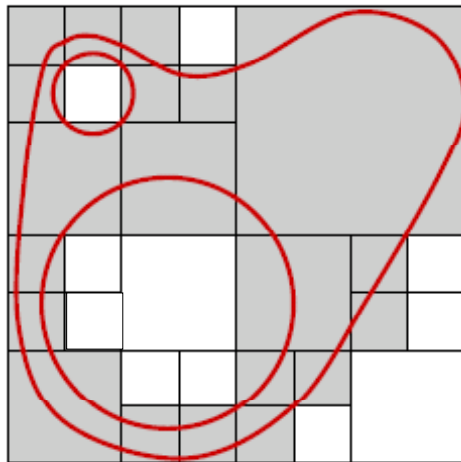
Representación de partición espacial: Voxels

- Problemas:
 - No existe el concepto de **ocupación parcial**, por lo tanto los sólidos con superficies curvas, sólo pueden aproximarse.
 - Sólo se pueden representar con exactitud los objetos cuyas caras son paralelas a los lados del cubo y cuyos vértices corresponden a la malla.
 - Requiere una gran cantidad de memoria: necesita N^3 celdas ocupadas para representar un objeto con una resolución de N elementos de volumen en cada una de las tres dimensiones espaciales.



Representación de partición espacial: árboles octales (octrees)

- Árboles octales: (octree):
 - Variante jerárquica de la enumeración de ocupación espacial, diseñada para optimizar los requisitos de almacenamiento de este método.
 - Un árbol octal se forma dividiendo sucesivamente una celda tridimensional en las tres direcciones (X, Y, Z) para formar octantes



Quadtree (equivalente al octree en 2D)



Representación de partición espacial: árboles octales

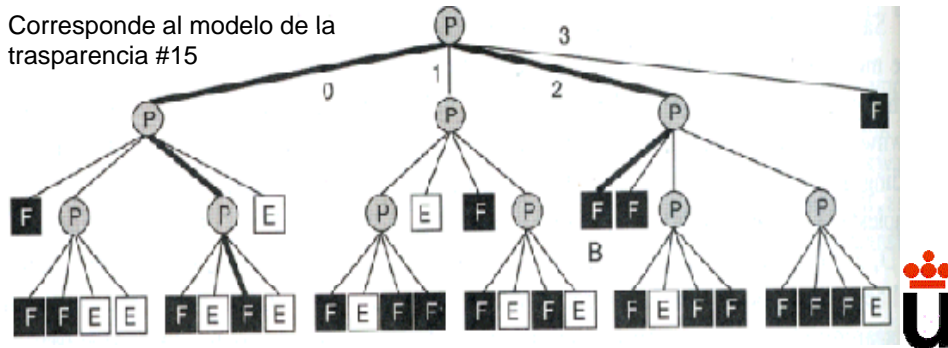
- Cada octante puede estar **lleno**, **parcialmente lleno**, o **vacío**
- Un octante parcialmente lleno se subdivide recursivamente en suboctantes si cumple algún criterio (no hay un criterio fijo).
- Este proceso de división continúa hasta que todos los octantes sean homogéneos o hasta alcanzar una profundidad límite previamente determinada.
- Cuando ocho octantes hermanos son homogéneos, se eliminan y su padre se reemplaza por un nodo totalmente lleno o vacío.
- Cualquier nodo parcialmente lleno en la profundidad límite se clasifica como lleno, con lo que tampoco en este caso existe el concepto de ocupación parcial.



Representación de partición espacial: árboles octales

- Representación con árbol de octantes parcialmente llenos en los nodos internos y llenos o vacíos en la hojas
- Se ha relajando el criterio para representar un nodo como homogéneo (que todos los subcuadrantes/suboctantes estén llenos o parcialmente llenos), la representación será más compacta aunque menos precisa.

Corresponde al modelo de la transparencia #15



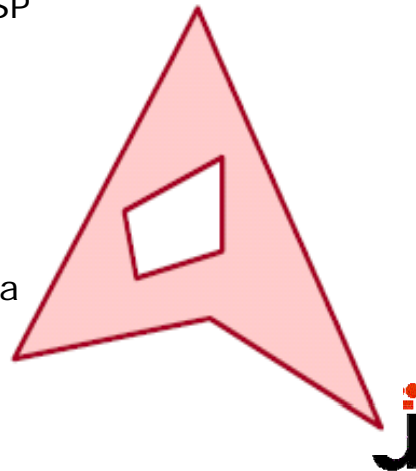
Representación de partición espacial: árboles BSP

- Árboles binarios de partición del espacio (BSP):
 - Dividen recursivamente el espacio en pares de subespacios, cada uno separado por un plano de orientación y posición arbitrarias.
 - Cada nodo interno del árbol BSP está relacionado con un plano y tiene dos nodos hijos, uno para cada lado del plano.
 - Suponiendo que las normales apunten hacia afuera del objeto, el hijo de la izquierda está **detrás** o **dentro** del plano, mientras que el de la derecha se encuentra **delante** o **fuera** del plano.
 - Si se divide aún más el semiespacio a un lado del plano, entonces su hijo es la raíz de un subárbol; si por el contrario el semiespacio es homogéneo y no tenemos que seguir dividiendo, el nodo es una hoja, y representa una región que está completamente dentro o fuera del poliedro.
 - Estas regiones homogéneas se denominan celdas **dentro** y **fuera**.

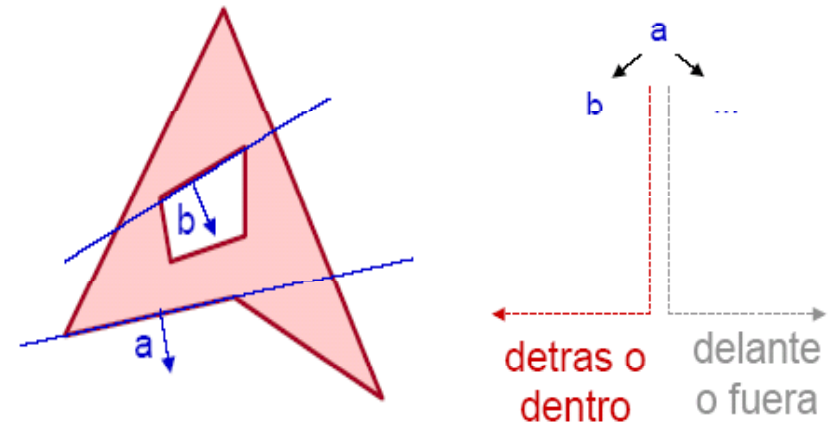


Representación de partición espacial: árboles BSP

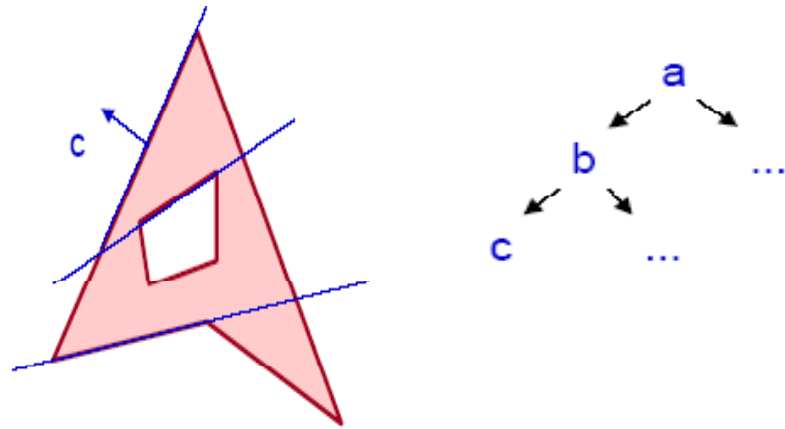
- El concepto de subdivisión que sirve de base a los árboles BSP es independiente de la dimensión.
- Sencillo y elegante.
- Representación poco compacta.
- Es posible desarrollar un álgebra booleana cerrada para árboles BSP tridimensionales.



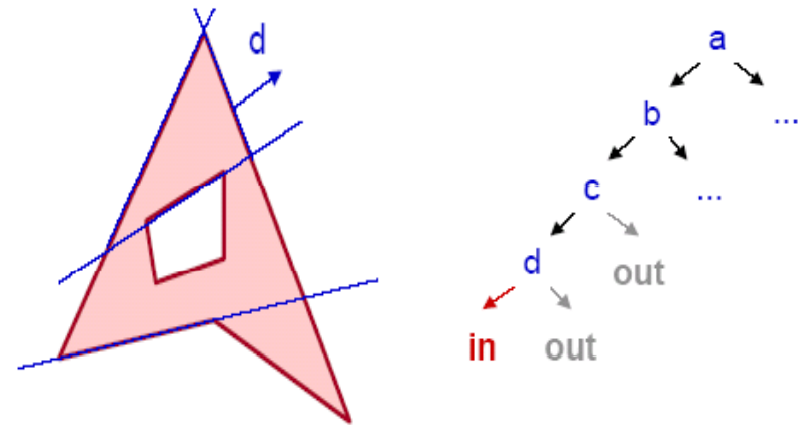
Representación de partición espacial: árboles BSP



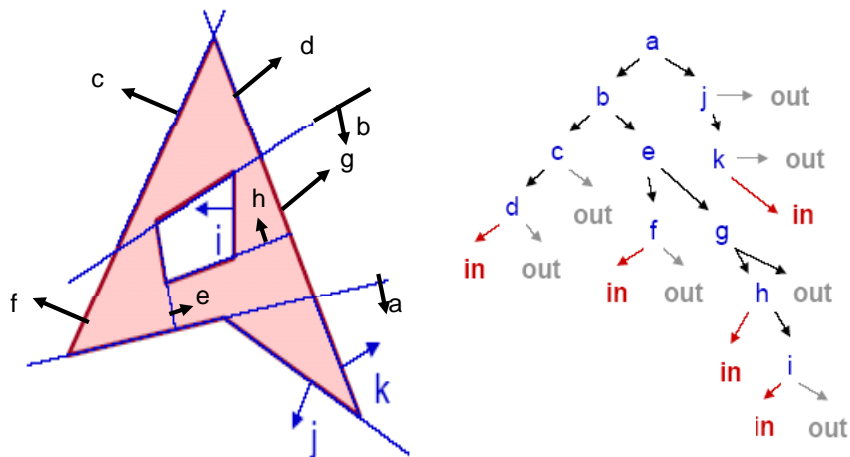
Representación de partición espacial: árboles BSP



Representación de partición espacial: árboles BSP

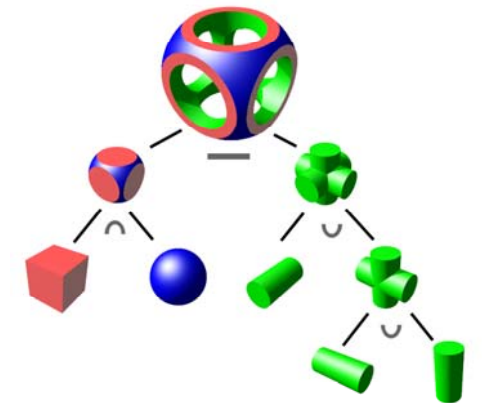


Representación de partición espacial: árboles BSP



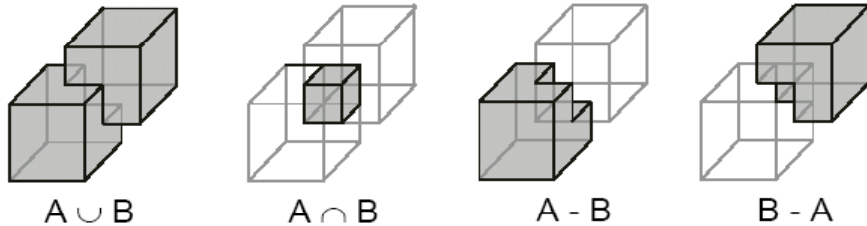
2.6 Geometría sólida constructiva: CSG

- Las primitivas simples se combinan a través de operadores booleanos de conjunto.
- Un objeto se almacena como un árbol binario:
 - Los nodos internos almacenan operadores booleanos o geométricos.
 - Las hojas almacenan primitivas simples.
- Las aristas del árbol deben estar ordenadas.



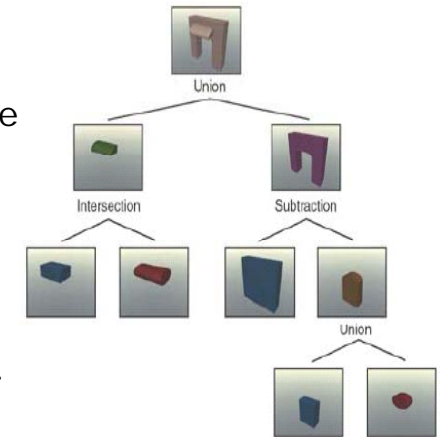
Geometría sólida constructiva: CSG

- Operaciones Booleanas con sólidos:
 - Combinan sólidos como conjuntos de puntos.
 - Pueden generar sólidos equivalentes de formas diferentes (no siempre hay una representación única).



Geometría sólida constructiva: CSG

- Gran capacidad de editar modelos a través de la eliminación, adición, reemplazo y modificación de subárboles.
- Almacenamiento compacto.
- Permite calcular con sencillez las propiedades físicas de los objetos.
- Resuelve con facilidad las interacciones entre objetos.
- Maneja de igual forma superficies curvas y poliédricas.



Geometría sólida constructiva: CSG

- Lento en la generación de escenas complejas (excepto traza de rayos: adaptado a este tipo de representación)
- Forma natural e intuitiva para el diseño de un gran conjunto de objetos (orientado al usuario)



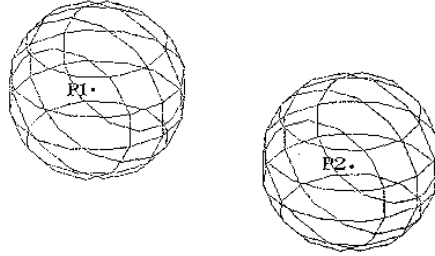
2.7 Modelos basados en funciones implícitas

- En la naturaleza existen muchos objetos con apariencia redondeada
- En muchos casos, lo importante es la forma global y no las dimensiones exactas
- Es necesario facilitar la generación y edición de este tipo de objetos
- Los modelos basados en funciones implícitas son un posible planteamiento, usado desde hace algunos años
- Muy usado, por ejemplo, para extraer la superficie de un líquido en simulación de fluidos, o para definir una malla 3D a partir de imágenes médicas (p.ej. algoritmo *marching cubes*)



Modelos basados en funciones implícitas

- Un objeto 3D se modela como una **isosuperficie** de un campo escalar que es generado por primitivas puntuales.
- El valor del campo creado por una primitiva P_i vendrá dado por una función de \mathbb{R}^3 en \mathbb{R} de dicho punto.



$$V_i(x, y, z) = f(x, y, z)$$



Modelos basados en funciones implícitas

- Combinando primitivas se suma el campo generado por cada una de ellas, y se pueden modelar formas complejas.

$$V(x, y, z) = \sum_{i=1}^N V_i(x, y, z)$$

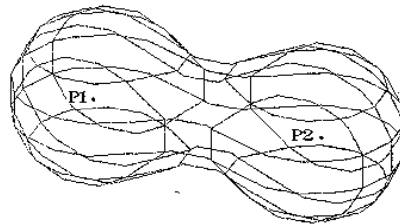
- Definiendo un color para cada primitiva (C_i) se puede calcular su valor en un punto con una simple ponderación (existen otros planteamientos posibles, por ejemplo basados en la conversión a otro tipo de modelo):

$$C(x, y, z) = \frac{1}{V(x, y, z)} \sum_{i=1}^N C_i V_i(x, y, z)$$



Modelos basados en funciones implícitas

- La suma de campos en el espacio genera una superficie deformada por la atracción de una esfera sobre la otra.
- La interacción entre las primitivas consigue deformaciones locales, con transiciones mucho más suaves.
- Plantea problemas la aplicación de texturas.

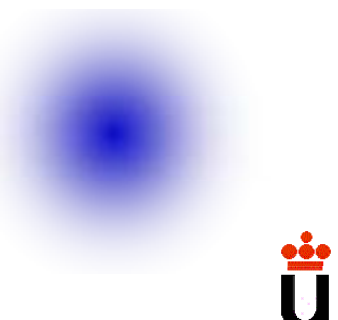


Modelos basados en funciones implícitas

- a : afecta a la caída exponencial del campo
- b : limita el valor que puede alcanzar el campo, en cualquier punto..
- $g(x, y, z)$ es el cuadrado de la distancia del punto a la primitiva y determina la forma de la primitiva si está aislada.
- Si el parámetro b es negativo la primitiva produciría una erosión sobre las adyacentes.
- El campo decae muy rápidamente con la distancia.

Bloppy model

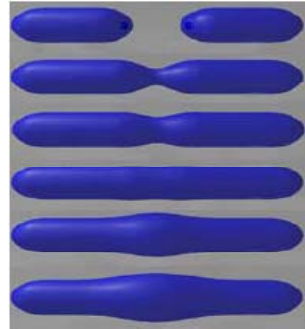
$$V_i(x, y, z) = b_i * \exp(-a_i g_i(x, y, z))$$



Modelos basados en funciones implícitas

- El campo tiene un alcance finito (depende de la distancia al cuadrado).
- La caída del campo es más suave. Por lo tanto, una primitiva tendrá más influencia en zonas más alejadas.

Soft objects



Modelos basados en funciones implícitas

- Admite diferentes funciones de campo.
- Distintas funciones influyen en las prestaciones del modelo:
 - Tiempo de cómputo
 - Calidad del objeto
 - Proceso de modelado.

Metaballs



3. Comparación de representaciones

- **Precisión:**
 - Los métodos de partición de espacio y representación poligonales de fronteras sólo producen aproximaciones de un gran número de objetos. Sin embargo, la resolución necesaria para producir gráficos de alta calidad, puede ser demasiado elevada para la práctica.
 - Por lo tanto, los sistemas que permiten utilizar gráficos de alta calidad, generalmente, emplean CSG con primitivas no poliédricas y representaciones de fronteras que permitan superficies curvas.
 - La generación por primitivas también puede crear imágenes de alta calidad, pero no permite combinar dos objetos sencillos utilizando operadores booleanos.
 - Por último, los campos escalares pueden generar imágenes de muy alta calidad para cierto tipo de objetos, si bien dependen de una selección adecuada de la función de campo.



3. Comparación de representaciones

- **Dominio:**
 - El dominio de los objetos que se pueden representar con la generación por primitivas y los desplazamientos es muy limitado.
 - En cambio los métodos de partición espacial pueden representar cualquier sólido, aunque muchas veces sólo como aproximaciones. Si se ofrecen otros tipos de caras y aristas además de los polígonos acotados por líneas rectas, es posible utilizar la representación de fronteras para representar una gama muy amplia de objetos.
 - Los *metaballs* pueden representar también cualquier tipo de sólido si bien la dificultad de modelarlo puede ser grande, y su aplicación se restringe a los modelos físicos ya comentados.



3. Comparación de representaciones

- **Unicidad:**
 - Los únicos métodos que garantizan la unicidad son los árboles de octantes y la enumeración de ocupación espacial.
 - La generación por primitivas no garantiza la unicidad, pero se puede asegurar si se elige con cuidado el conjunto de primitivas.
 - El resto de los métodos no garantizan esta propiedad.
- **Validez:**
 - Las representaciones de fronteras son las más difíciles de validar.
 - Un árbol BSP representa un conjunto espacial válido, aunque no necesariamente un sólido acotado.
 - Sólo hay que efectuar una sencilla revisión sintáctica local para validar un árbol CSG o un árbol de octantes, y no se requiere ninguna verificación para la enumeración de ocupación espacial.
 - En *metaballs*, si se eligen correctamente las funciones que generan campos de forma que sean continuas, no hay problemas de validez.



3. Comparación de representaciones

- **Cerradura:**
 - Los objetos creados con la generación por primitivas no se pueden combinar y los desplazamientos simples no son cerrados, al aplicarse operaciones booleanas de conjuntos. Por lo tanto, ninguno de estos dos métodos se emplea como representación interna en sistemas de modelado.
 - Algunas representaciones de fronteras pueden presentar problemas de cierre con operaciones booleanas.
- **Almacenamiento y eficiencia:** Los métodos de representación se clasifican en función de su capacidad para producir modelos **evaluados** o **no evaluados**.
 - Los modelos no evaluados contienen información que debe procesarse para efectuar operaciones básicas, como la determinación de la forma del objeto.
 - En lo que respecta a la utilización de operaciones booleanas, la CSG crea modelos no evaluados, ya que cada vez que se realizan cálculos hay que recorrer el árbol y evaluar las expresiones.



3. Comparación de representaciones

- Como resumen se puede decir que no existe una representación exacta y óptima que modele de forma adecuada objetos sólidos en tres dimensiones.
- Cada una tiene ciertas ventajas e inconvenientes, y cada una es más adecuada para un tipo específico de aplicaciones.
- Algunos sistemas comerciales emplean varias de estas representaciones simultáneamente, ya que algunas operaciones concretas son más eficientes en unos modelos que en otros.

