

# Universidade Estadual do Norte do Paraná

Computação Gráfica

Prof. Bruno Miguel

## Transformações de Pontos e Objetos

Translação

Escala

Rotação

Reflexão

Cisalhamento

## Modelagem Geométrica

### FORMAS DE ARMAZENAMENTO

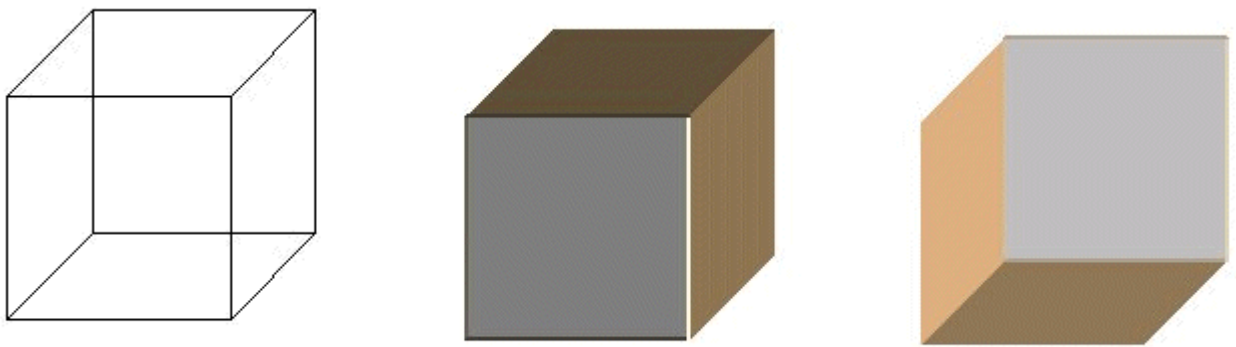
#### Representação Aramada ou por *"wire frame"*

Nesta representação os objetos são descritos por um conjunto de arestas que definem as bordas do objeto. O método nada mais é do que uma extensão 3D do método de representação de objetos 2D por arestas.

A principal vantagem desta técnica é a sua velocidade na exibição dos modelos pois é necessário apenas exibir um conjunto de linhas.

Porém, há sérios inconvenientes no uso da representação aramada:

- ela gera uma representação "ambígua". Ou seja, quando o modelo é exibido pode dar margem a mais de uma interpretação. A figura 1 exemplifica o problema. Em "a" tem-se a representação aramada e em "b" e "c", duas possíveis interpretações. O problema não reside propriamente no fato de que a simples exibição das linhas gera ambigüidades, mas sim, na constatação de que o modelo não fornece informações suficientes para que estas sejam eliminadas (no exemplo seria necessário remover as arestas da parte traseira do objeto);
- é bastante difícil, e em alguns casos impossível, realizar certas operações como a determinação de massa, volume, inclusão ou não de pontos, etc.



**Figura 1 - Ambigüidades da representação aramada**

---

## **Representação por faces(ou superfícies limitantes)**

Esta técnica consiste em definir um modelo através de um conjunto de polígonos que delimitam uma região fechada do espaço. Esta região define o interior do modelo. Aos polígonos que limitam a região dá-se o nome de **FACES**. O objeto formado por esta técnica é normalmente chamado de **POLIEDRO**, ou seja, composto de muitos **DIEDROS** (diedro = semi-espaço).

Esta técnica também pode ser considerada uma extensão(mais inteligente) da modelagem 2D por arestas. Como naquele caso, onde há, pelo menos, duas formas de armazenamento das arestas, neste, é possível fazer uma analogia com as faces, que podem ser representadas de duas formas básicas:

- através de uma lista de vértices explícitos, na forma

**FACE: (x1, y1, z1)-(x2, y2, z2)- .... - (xn, yn, zn);**

- através de duas listas, numa onde é dada a topologia da face(apenas o número dos vértices que a compõem), na forma

**FACE: v1, v2, v3, ....., vn**

- e noutra onde é definida a geometria da face, com a explicitação das coordenadas de cada vértice do modelo, na forma

### **VÉRTICES:**

**1 - (x1, y1, z1)**

**2 - (x2, y2, z2)**

**.... ....**

**n - (xn, yn, zn)**

---

## Representação por Enumeração de Ocupação Espacial

Também conhecidos como **Modelos de Subdivisão(ou Decomposição) do Espaço** esta classe de formas de armazenagem decompõem o sólido em "pedaços".

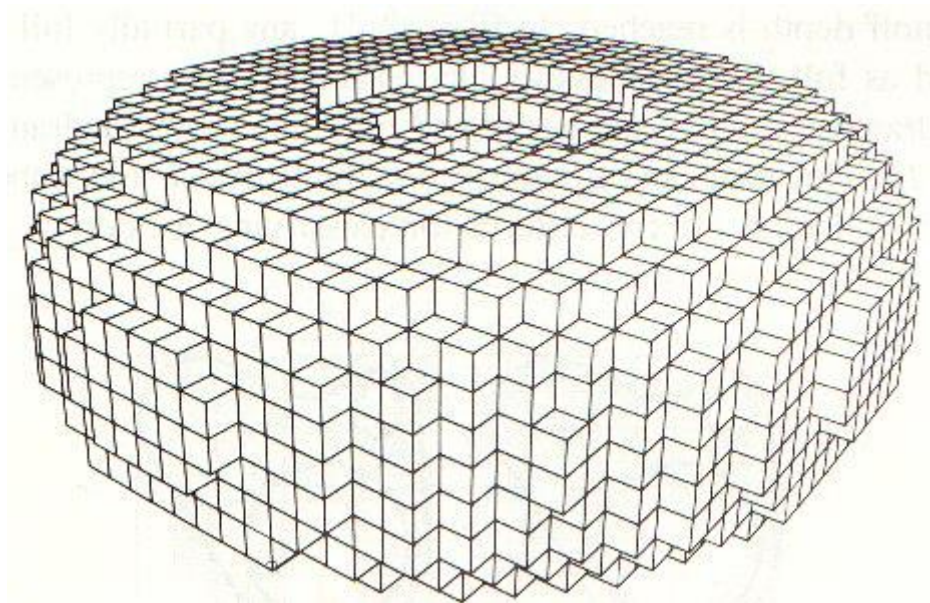
### Decomposição em células ou partes

- Um sólido é visto como uma coleção de partes *mais simples*
- Problema é saber o que significa uma parte *mais simples*. Na modelagem de um computador, por exemplo, o teclado pode ser uma das partes e o monitor outra.

### Enumeração Exaustiva/Enumeração de Ocupação Espacial

A idéia é dividir o espaço em regiões e definir o objeto através das regiões que ele ocupa neste espaço. Esta técnica é bastante útil quando se deseja calcular propriedades de massa, pois basta saber, por exemplo o volume de uma das partes em que o espaço foi dividido e multiplicar este valor pelo número de divisões ocupadas pelo objeto.

- O espaço é subdividido em cubos formando um GRADE TRIDIMENSIONAL
- A cada um destes cubos dá-se o nome de **voxel**
- Codifica-se um sólido determinando quais os **voxels** pertencem a ele



**Figura 2 - Exemplo de objeto descrito por Voxels**

### Vantagens

- É fácil determinar se um dado ponto pertence ou não ao sólido basta verificar se o ponto pertence a algum dos *voxels*
- É fácil determinar se dois objetos se interferem(se tocam)
- Facilita a realização de operações de união, intersecção e diferença entre sólidos

### Desvantagem

- Uma representação detalhada necessita de muita memória

## Representação por *Octrees* e *Quadtrees*

[Clique aqui para copiar um programa que armazena objetos por \*quadrees\*](#)

(Autores: Patrícia Zottis e Rodrigo Fehse; Alterações: Leonardo Langie)

Considerada um caso particular da Enumeração de Ocupação Espacial, a técnica "**representação por OCTREE**" (ou árvore com 8 filhos) envolve o objeto, que em seguida é dividido em 8 cubos menores de igual tamanho(**OCTANTES**). Cada um destes é então classificado em:

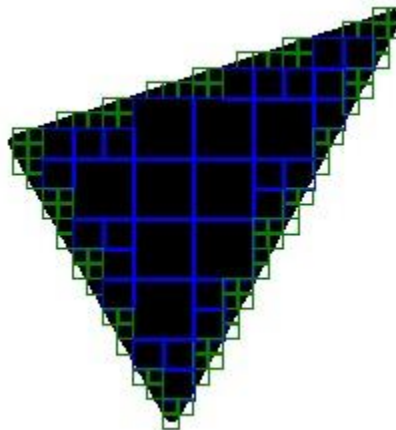
- **Cheio**, caso o objeto ocupe todo o cubo;
- **Vazio**, caso o objeto não ocupe nenhuma parte do cubo;
- **Cheio-Vazio**, caso o objeto ocupe parte do cubo;

Quando um octante for classificado em "**Cheio-Vazio**" ele é novamente dividido em 8 partes iguais e o processo de classificação é refeito para as novas partes. Este algoritmo repete-se até que só hajam cubos com as duas primeiras classes.

Esta forma de armazenamento é um caso especial de Enumeração Espacial. Neste caso, os **voxels** passam a ser cubos de dimensões variáveis.

Para o armazenamento de objetos 2D, usa-se as **QUADTREES**. Nelas divide-se o plano onde está o objeto em 4 partes iguais e classifica-se cada parte da mesma forma que com *octrees*.

Em geral armazena-se a estrutura de *octrees* e *quadtrees* em forma de árvore.



**Figura 3 - Exemplo de objeto descrito por Quadtree**

### Vantagens

- Tem as mesmas vantagens da Enumeração Espacial
- Permite uma representação mais detalhada com um gasto menor de memória

### Desvantagem

- São mais trabalhosas para manipular

---

## TÉCNICAS DE MODELAGEM

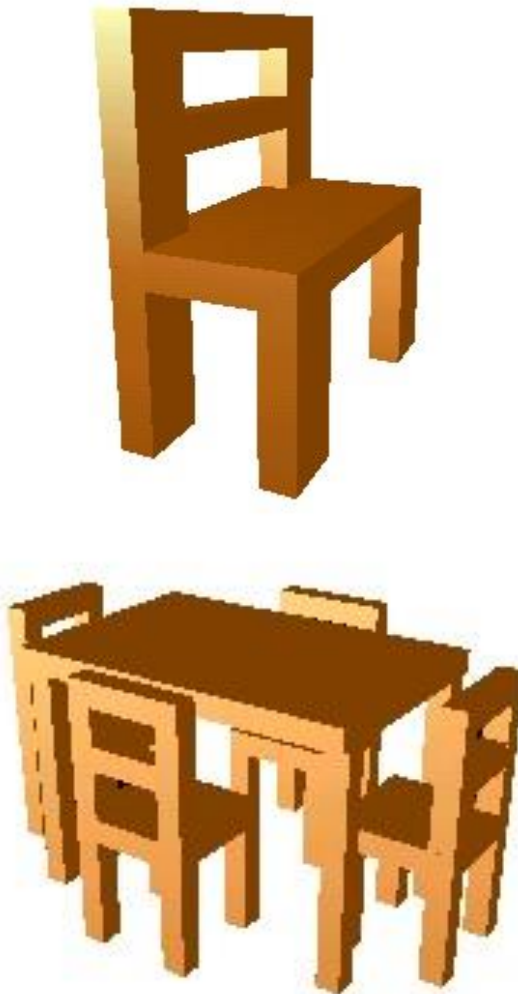
### **Instanciamento de Primitivas***(justaposição de sólidos primitivos)*

Esta técnica de modelagem cria novos objetos através do posicionamento de objetos por transformações geométricas

Outra forma de criar objetos com esta técnica é o uso de **Primitivas Parametrizáveis**. Com isto pode-se criar, por exemplo, um conjunto de peças mecânicas ou roupas, com apenas alguns comandos.

[Clique aqui para copiar um programa de modelagem por justaposição](#). (autores: Afonso Sales e Michele Cardoso)

Na figura a seguir pode-se observar uma cadeira que foi modelada pela justaposição de paralelepípedos e uma sala modelada pela justaposição de cadeiras e de uma mesa.



**Figura 4 - Exemplo de objetos modelados por justaposição**

---

### **Geometria Sólida Construtiva (*CSG-Contrutive Solid Geometry*)**

Método de CSG consiste em construir um objeto a partir da combinação operatória (união, intersecção e diferença) de dois ou mais sólidos. A figura 5 exemplifica o uso desta técnica. No exemplo pode-se observar a realização de uma diferença entre um retângulo e uma elipse.

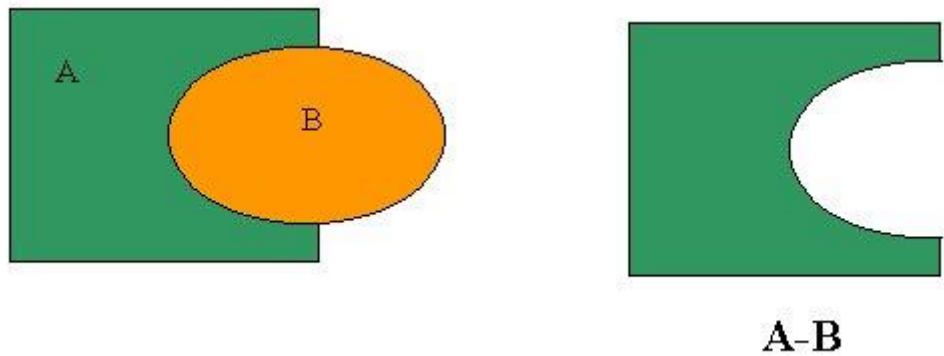


Figura 5 - Uso da técnica de CSG

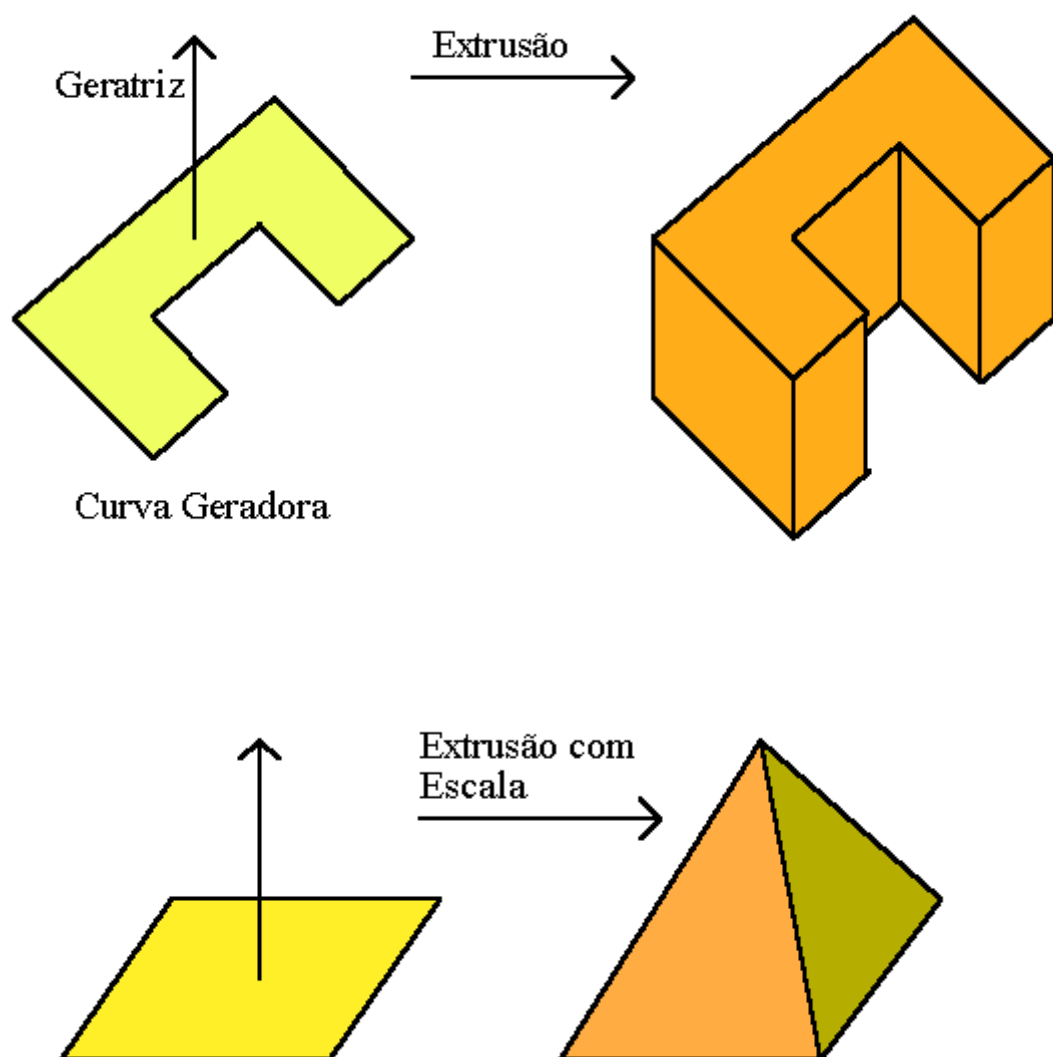
---

## Modelagem por Varredura(sweep)

A representação por varredura cria objetos baseada na noção de que uma curva C1 quando deslocada no espaço ao longo de uma trajetória dada por uma outra curva C2 descreve uma superfície que pode ser usada para definir um sólido. À curva C1 dá-se o nome de **CONTORNO** ou **GERATRIZ** e à C2, o nome de **CAMINHO** OU **DIRETRIZ**.

### Varredura Translacional(Extrusão)

Um objeto "O" definido por varredura translacional é obtido pela translação de uma curva C ao longo de um vetor V, por um distância D. A figura 6 exemplifica a criação de dois objetos por esta técnica. A varredura translacional de um retângulo gera um paralelepípedo, de uma circunferência gera um cilindro.



**Figura 6 - Criação de objetos por varredura translacional**

### **Varredura Rotacional**

Neste tipo de modelagem por varredura a superfície do objeto é descrita por uma curva que gira em torno de um eixo. Na figura 7 temos, à esquerda, uma curva C e à direita o objeto gerado pela rotação de C em torno do eixo Y. [Clique aqui para obter um programa exemplo de Sweep rotacional](#) (Autor: Rafael Prikladinik).





**Figura 7 - Criação de objeto por varredura rotacional**

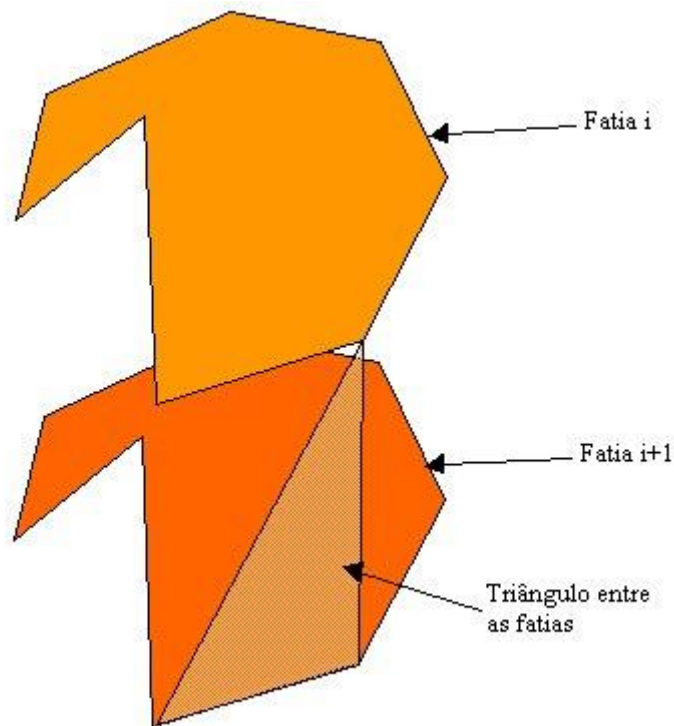
---

## **Modelagem por Seções Transversais**

Esta técnica permite gerar sólidos através de cortes. A idéia básica é obter seções transversais do objeto que se deseja modelar. A partir destas seções, use-se seus pontos através de triângulos. As fatias podem ser obtidas pelo corte do objeto ou por processos como tomografia.

Toma-se um eixo (segmento de reta)

- Corta-se o sólido em fatias perpendiculares ao eixo
- Digitaliza-se o contorno de cada fatia
- Une-se as fatias com triângulos



**Figura 8 - Criação de objeto por seções planas**

---

Alguns LINKS:

APIs para desenvolvimento de Ambientes 3D

**WEBGL:** <https://threejs.org> Exemplos: <https://threejs.org/examples/>

**OPENGL:** <http://opengl.org>

**Software para modelagem:**

- Sketchup (<http://sketchup.com>)

-Blender (<http://blender.com>)

**Engines (Ambientes pré-programados)**

Unreal-Engine (<https://www.unrealengine.com/>)

Ogre (<https://www.ogre3d.org/>)

J-Monkey engine (<http://jmonkeyengine.org/>)

Unity (<http://www.unity.com/>)

**Projeto de Jogo 3D desenvolvido por alunos da UNICAMP:**

<http://www.dca.fee.unicamp.br/courses/IA725/1s2006/projeto/g5/v3/index.html>

**Referências:**

PINHO, Modelagem de Sólidos, disponível em

<http://www.inf.pucrs.br/~pinho/CG/Aulas/Modelagem/Modelagem3D.htm> – Acessado em 23/06/2018

NUNES, E.O.; Modelagem Geométrica e Sweeping, disponível em:

<http://www2.ic.uff.br/~aconci/sweeping.html> – Acessado em 23/06/2018