

<http://computacaografica.ic.uff.br/conteudocap4.html>

REPRESENTAÇÃO DE DADOS EM CG

MODELAGEM E ESTRUTURA DE DADOS

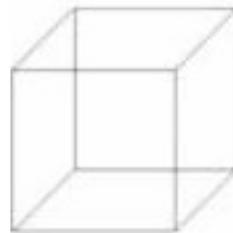
Aula 5 – UFF – 2018/2

No site essa aula esta como:

www.ic.uff.br/~aconci/CG-Aula3-2017.pdf

FORMAS DE REPRESENTAÇÃO

- **Representação Aramada (Wire Frame):**
 - representação ambígua com margem para várias interpretações;
 - dificuldade de realizar certas operações como a determinação de massa ou volume. e
 - não tem como garantir que o objeto desenhado seja um sólido válido,



Representação por Faces (ou Superfícies Limitantes)

- Essas superfícies são supostas fechadas e orientáveis.
- Orientáveis = significa que é possível distinguir entre dois lados da superfície, de modo que um esteja no interior e o outro no exterior do sólido.

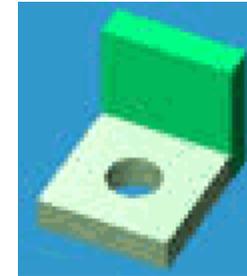
Formula ou lei de Euler-Poincaré:

$$V-A+F-H=2(C-G)$$

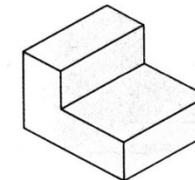
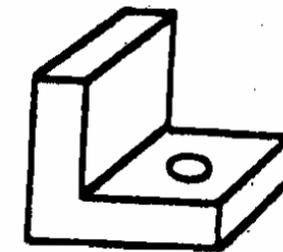
H= loops de faces fechadas;

C= numero de partes separadas do objeto

G= numero de buracos (genus)



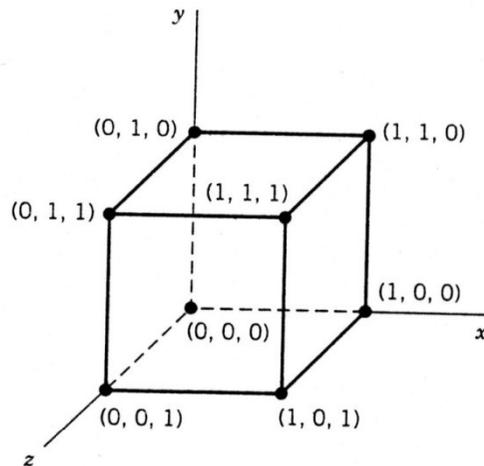
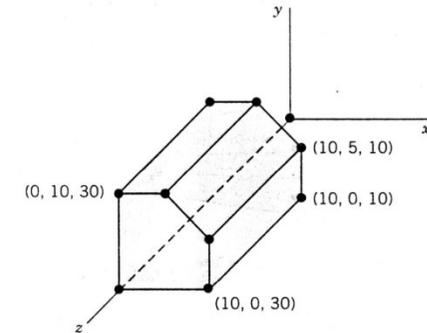
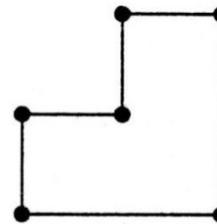
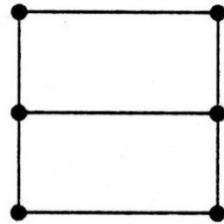
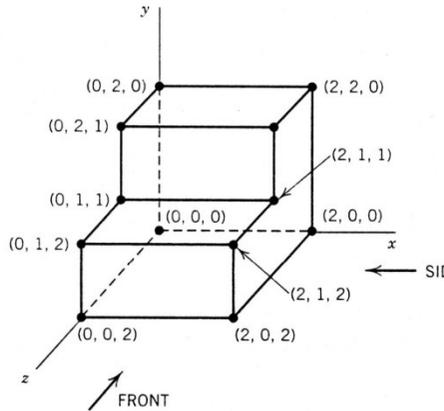
Boundary representation
B-rep



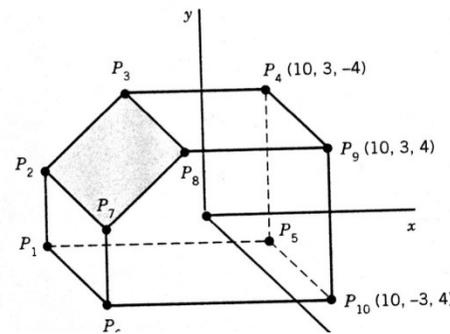
Descrição da:

- **topologia** e a **geometria** das faces;
- relações entre os elementos;
- posições dos elementos no espaço, e sua forma geométrica (semi-reta, arco de círculo, etc)

Geometria x topologia



$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$



$$[P] = \begin{bmatrix} -10 & -3 & -4 & 1 \\ -10 & 1 & -4 & 1 \\ -8.5 & 3 & -4 & 1 \\ 10 & 3 & -4 & 1 \\ 10 & -3 & -4 & 1 \\ -10 & -3 & 4 & 1 \\ -10 & 1 & 4 & 1 \\ -8.5 & 3 & 4 & 1 \\ 10 & 3 & 4 & 1 \\ 10 & -3 & 4 & 1 \end{bmatrix}$$

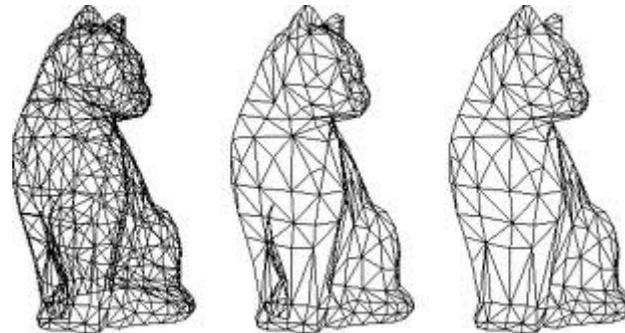
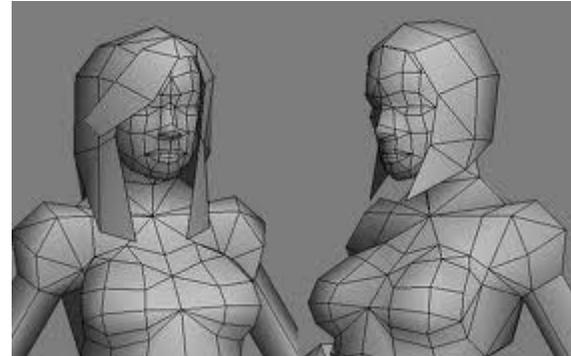
Representação dos limites do sólido

- Boundary Representation – Brep
- É a forma mais usada
- **Nela toda a topologia é considerada para garantir que o objeto seja realizável e continue realizável após as operações que serão realizadas nele.**
- **A topologia deve ser validada não só a geometria gerada** (Equação de Euler)

estrutura de dados do objeto.

Data structure

- **Polygon-based (Face list)**
- **Vertex-based**
- **Edge-based**



Estrutura de dados baseada Faces e Vértice

vertex *coordinates*

v_1 $x_1 y_1 z_1$

v_2 $x_2 y_2 z_2$

v_3 $x_3 y_3 z_3$

v_4 $x_4 y_4 z_4$

v_5 $x_5 y_5 z_5$

v_6 $x_6 y_6 z_6$

v_7 $x_7 y_7 z_7$

v_8 $x_8 y_8 z_8$

face *vertices*

f_1 $v_1 v_2 v_3 v_4$

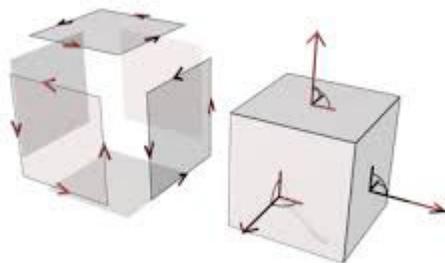
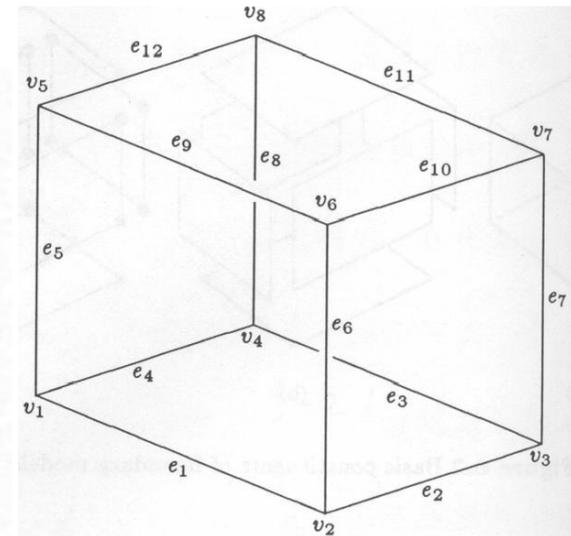
f_2 $v_6 v_2 v_1 v_5$

f_3 $v_7 v_3 v_2 v_6$

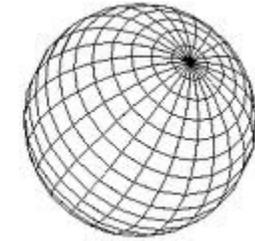
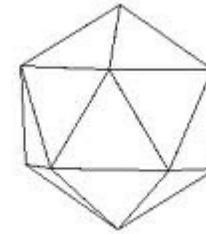
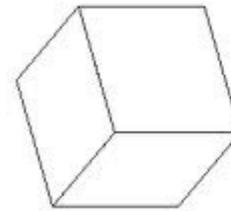
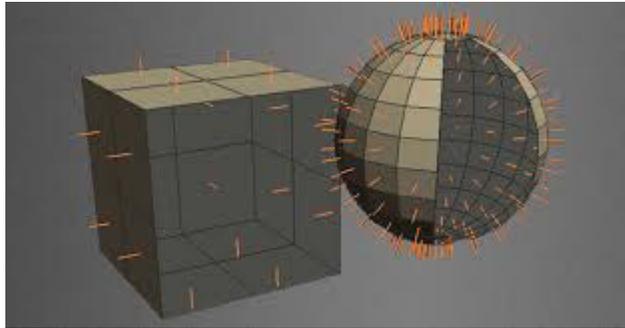
f_4 $v_8 v_4 v_3 v_7$

f_5 $v_5 v_1 v_4 v_8$

f_6 $v_8 v_7 v_6 v_5$

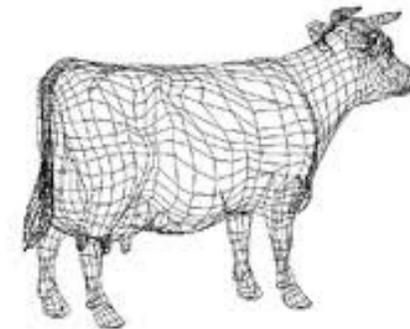
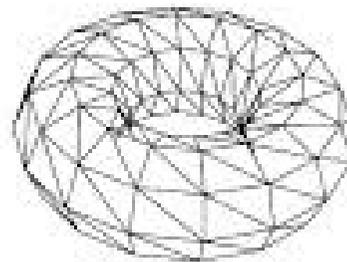


os vértices limites das faces devem ser descritos **sempre no mesmo sentido horário** (ou anti-horário) do exterior do objeto, **para todas as faces**.



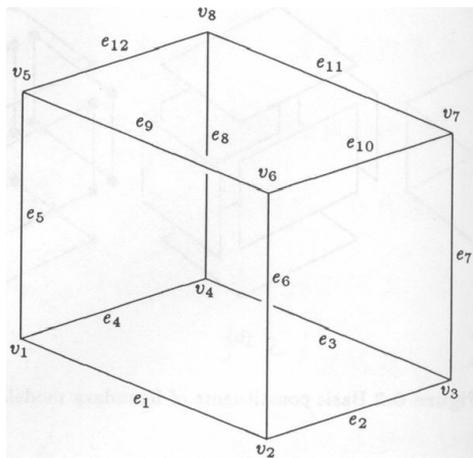
Estrutura de Dados Baseada em Arestas ou Lados

Na estrutura de dados baseada em arestas além das listas de coordenadas de vértices e definição das faces, tem-se uma lista que identifica cada aresta e seus vértices limitantes.



Baseada em lados (edges)

- Lados são considerados orientados.
- Cada lado pertence a duas faces.
- Faces são consideradas orientadas, positivas se sua lista de lados apontar para fora se for no sentido horário

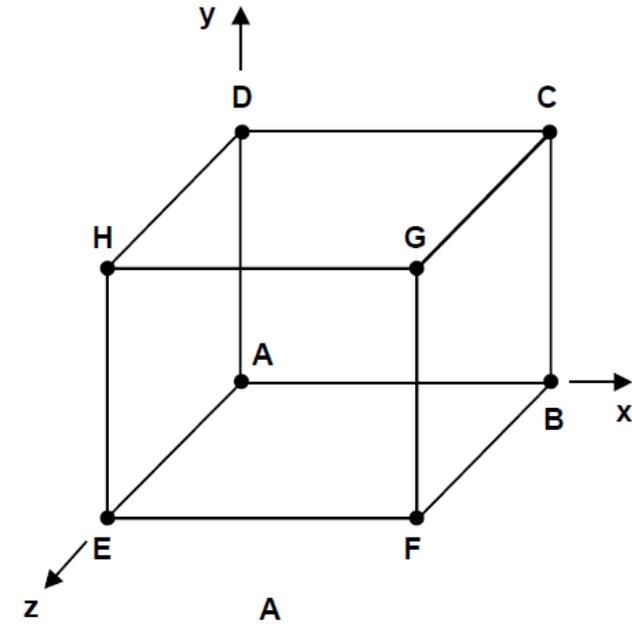


<i>edge</i>	<i>vertices</i>	<i>vertex</i>	<i>coordinates</i>	<i>face</i>	<i>edges</i>
e_1	$v_1 v_2$	v_1	$x_1 y_1 z_1$	f_1	$e_1 e_2 e_3 e_4$
e_2	$v_2 v_3$	v_2	$x_2 y_2 z_2$	f_2	$e_9 e_6 e_1 e_5$
e_3	$v_3 v_4$	v_3	$x_3 y_3 z_3$	f_3	$e_{10} e_7 e_2 e_6$
e_4	$v_4 v_1$	v_4	$x_4 y_4 z_4$	f_4	$e_{11} e_8 e_3 e_7$
e_5	$v_1 v_5$	v_5	$x_5 y_5 z_5$	f_5	$e_{12} e_5 e_4 e_8$
e_6	$v_2 v_6$	v_6	$x_6 y_6 z_6$	f_6	$e_{12} e_{11} e_{10} e_9$
e_7	$v_3 v_7$	v_7	$x_7 y_7 z_7$		
e_8	$v_4 v_8$	v_8	$x_8 y_8 z_8$		
e_9	$v_5 v_6$				
e_{10}	$v_6 v_7$				
e_{11}	$v_7 v_8$				
e_{12}	$v_8 v_5$				

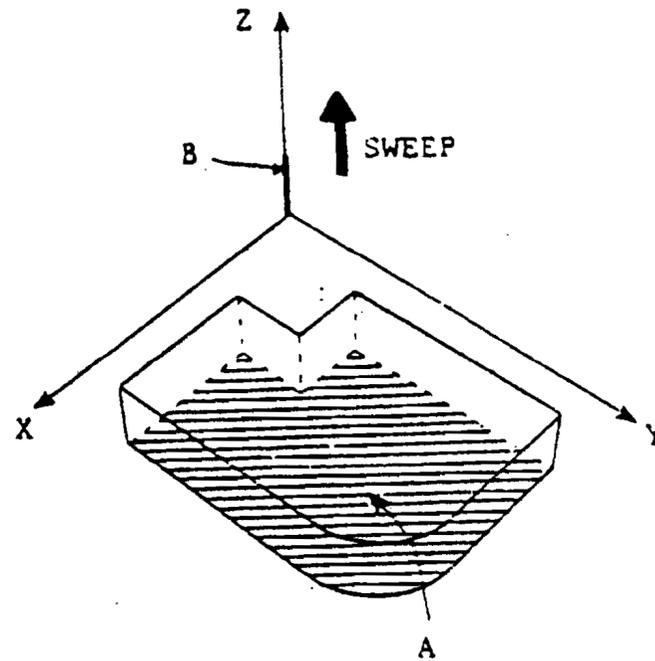
Vértices	Coordenadas
A	(0,0,0)
B	(1,0,0)
C	(1,1,0)
D	(0,1,0)
E	(0,0,1)
F	(1,0,1)
G	(1,1,1)
H	(0,1,1)

Aresta	Vértices
A1	EF
A2	FB
A3	BA
A4	AE
A5	EH
A6	FG
A7	BC
A8	AD
A9	HG
A10	GC
A11	CD
A12	DH

Faces	Arestas
F1	A1 A2 A3 A4
F2	A9 A6 A1 A5
F3	A6 A10 A7 A2
F4	A7 A11 A8 A3
F5	A12 A5 A4 A8
F6	A9 A12 A11 A10



Sweep : superficies 2D GERAM o OBJETO 3D



Translational sweeping.

Referencias

- D. F. Rogers, J. A. Adams. Mathematical Elements for Computer Graphics, 2dn Ed. , Mc Graw Hill, 1990
- E. Azevedo, A. Conci, [Computação Gráfica: teoria e prática](#), [Campus](#) ; - Rio de Janeiro, 2003
- J.D.Foley,A.van Dam,S.K.Feiner,J.F.Hughes. Computer Graphics- Principles and Practice, Addison-Wesley, Reading, 1990.

Trabalho 1 Implementação

Desenhe um objeto 2D formado pela união de retas, que correspondam aos números 0, 1 e 2 e as letras I, C, U, F, W e S (vetorial) em maiúsculas com alguma espessura.

Anime esse objeto de modo que ele passe a se deslocar em um plano 2D de uma posição x_1, y_1 inicial até uma posição x_2, y_2 . E ao mesmo tempo ele vá girando de 90 graus em relação a posição anterior.

Os números e as letras podem estar em qualquer cor.

Trabalho 1 Implementação – cont.

O trabalho pode ser feito em grupos de até 3 pessoas e em qualquer linguagem.

Cada grupo pode escolher uma letra ou 1 número para ser animado, mas 2 grupos não podem escolher o mesmo caractere.

Onde serão os pontos iniciais e finais serão definidos pela professora depois de vocês mostrarem a elas o desenho dele estático.

Esse desenho depois será transformado em 3D por sweep (a parte estática é para ser entregue até 12/09/2018).

Leonhard Euler

Fórmula ou lei de Euler:
 $V-E+F=2$

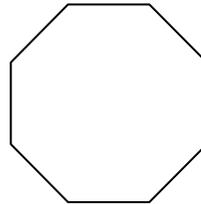


(1707-1783)

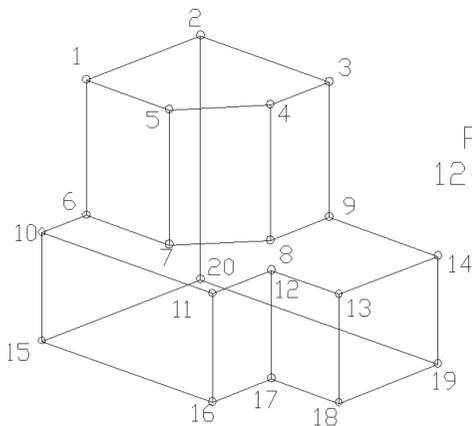
$$V=E=4 \quad F=2$$



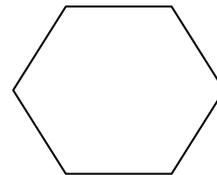
$$V=E=8 \quad F=2$$



$$\begin{aligned} V &= 20 \\ E &= 30 \\ F &= 12 \\ F-E+V &= 2 \\ 12-30+20 &= 2 \end{aligned}$$



$$V=E=6 \quad F=2$$



Euler's Law

