



Representação de Objetos Tridimensionais

Malhas Poligonais

Computação Gráfica
Maria Cristina F. de Oliveira
Rosane Minghim
2007-2018

Modelo

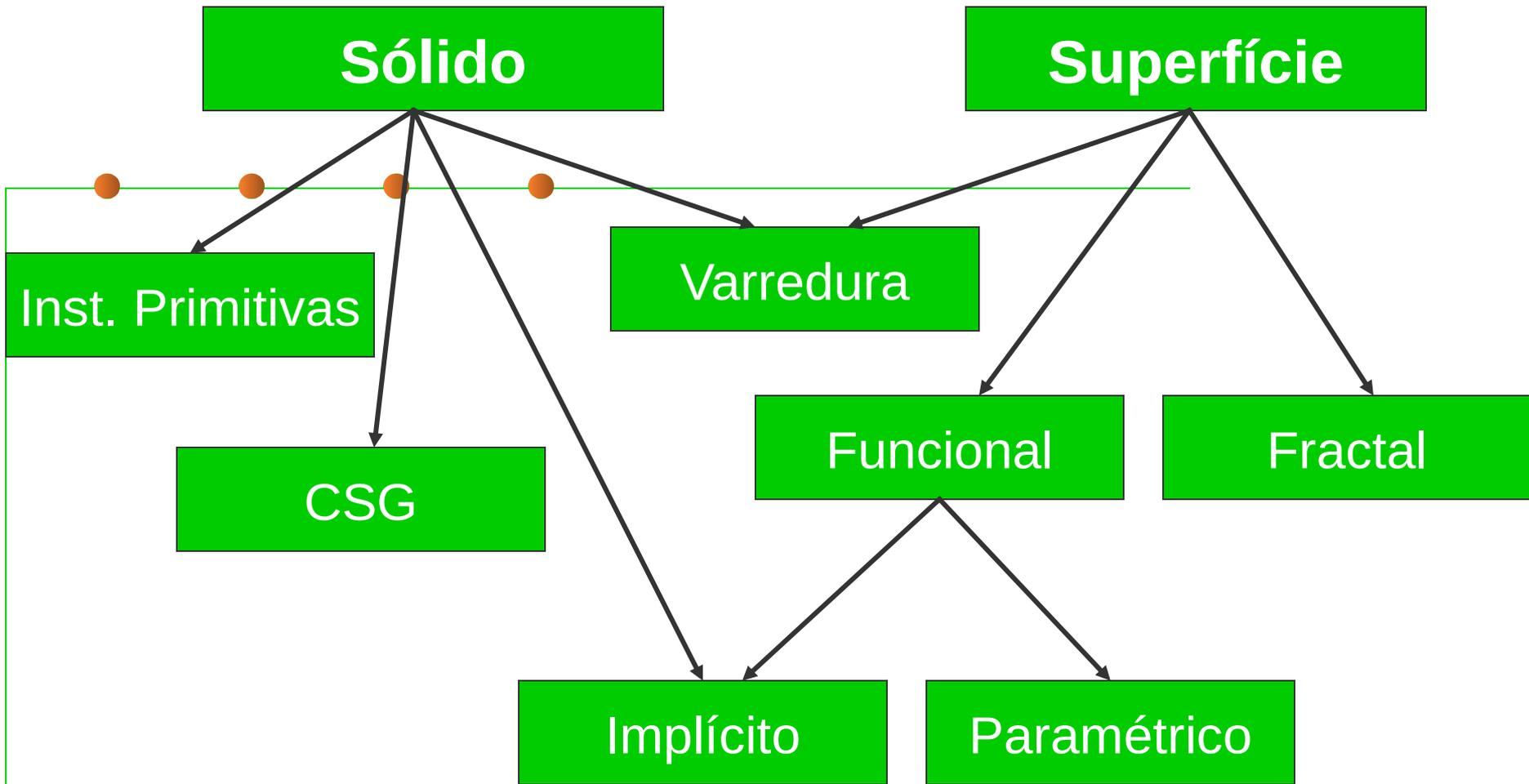
- Representação construída artificialmente para tornar mais fácil a observação/análise de um objeto/fenômeno
 - Nível de detalhe definido pelas aplicações que o utilizam
 - Problemas práticos em CG: modelos geométricos

Modelagem Geométrica

- Início dos anos 70
- Coleção de métodos usados para descrever a forma e outras características geométricas de um objeto, bem como para simular processos dinâmicos
- Sistema de modelagem geométrica: sistema computacional que permite a criação, modificação e acesso à representação de objetos por meio de modelos geométricos

Modelos Geométricos

- Cenas gráficas podem conter muitos tipos diferentes de objetos e materiais
 - Não existe uma maneira única capaz de descrever e representar todos os tipos de objetos
- Descrição vs. Representação
 - Descrição do objeto pelo usuário: processo de modelagem
 - Representação do objeto no sistema gráfico: como manter as informações necessárias para renderizar o objeto (e aplicar simulações, por ex.)



Sólido

Superfície

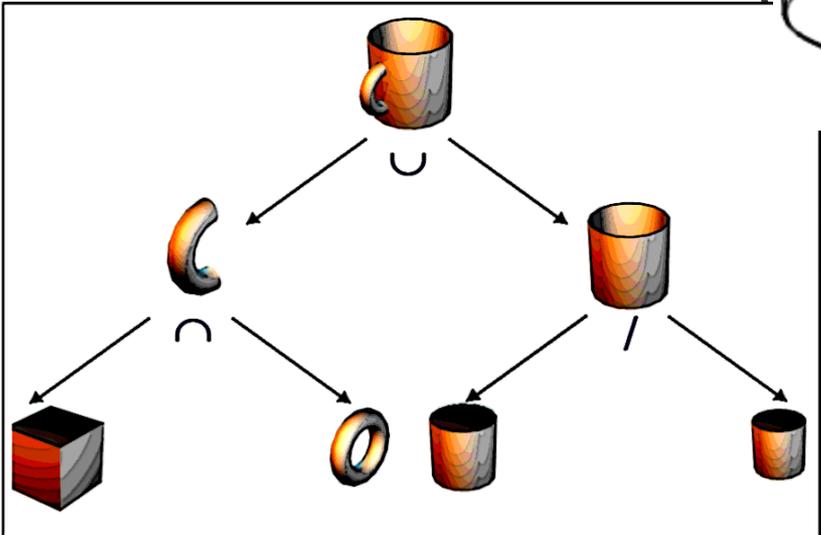
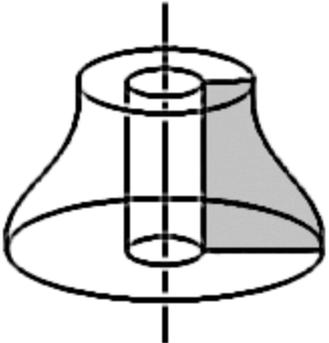
Inst. Primitivas

Varredura

CSG

Funcional

Fractal



Alícito

Para



Sólido

Superfície

Inst. Primitivas

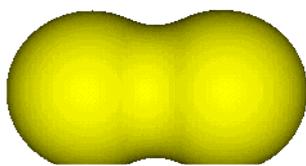
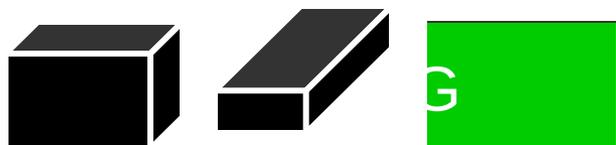
Varredura

Funcional

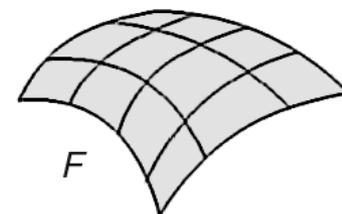
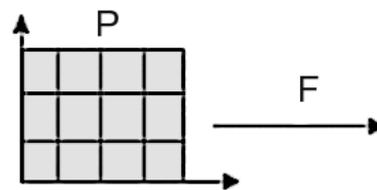
Fractal

Implícito

Paramétrico



$$f(x) = 0$$



Modelagem de Sólidos

- Ramo da M.G. que estuda técnicas para criar, manter e comunicar informação sobre a forma de objetos sólidos
 - envolve a criação e a manutenção de um modelo para futuro acesso e análise
 - permite formular e responder questões sobre propriedades volumétricas (volume, peso, momento de inércia, ...) e topológicas (conectividade, pertinência, ...)

Volume vs. Superfícies

- Objetos sólidos tridimensionais
 - Representados apenas pela sua fronteira
 - representações por fronteira: objeto 3D descrito como um conjunto de superfícies que separa o seu interior do meio externo (geralmente quadriláteros ou triângulos, ou superfícies paramétricas)
 - Superfície e conteúdo interno representados explicitamente
 - Representações por particionamento espacial descrevem propriedade interiores, particionando a região do espaço que contém o objeto em um conjunto de pequenos sólidos adjacentes não sobrepostos (geralmente cubos ou tetraedros)

Malhas Poligonais

- Atualmente existe uma enorme diversidade de técnicas e modelos em CG
- Vamos estudar inicialmente uma forma de representação por fronteira muito simples, adotada em muitos sistemas gráficos
 - objetos descritos por malhas poligonais que representam a sua superfície (fronteira)
 - Conjunto de vértices, arestas e faces planares (triângulos)
 - Representação adequada para 'rendering' por placas gráficas: objetos gráficos padrão

Malhas Poligonais

- Poliedros
 - Representação poligonal é exata
- Objetos em geral
 - descritos por superfícies curvas
 - decompostos (*tesselated*) para produzir uma representação poligonal aproximada

Malhas Poligonais

Exemplos

Fonte:

<http://www.bymath.com/studyguide/geo/geo21.htm>

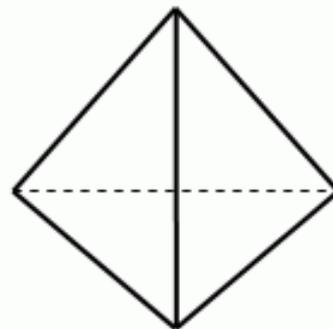


Fig. 99

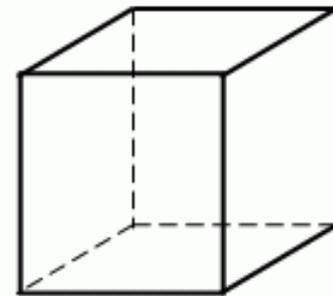


Fig. 100

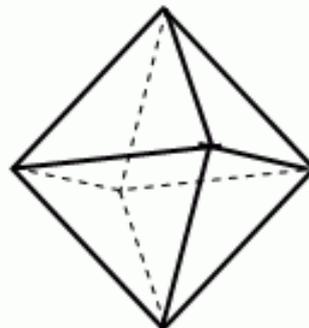


Fig. 101

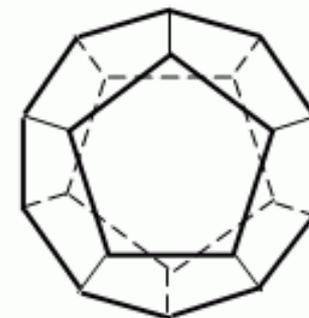


Fig. 102

Poliedros convexos regulares

tetraedro (4 faces, Fig.99)

hexaedro (cubo, 6 faces, Fig.100)

octaedro (8 faces, Fig.101)

dodecaedro (12 faces, Fig.102)

icosaedro (20 faces, Fig.103)

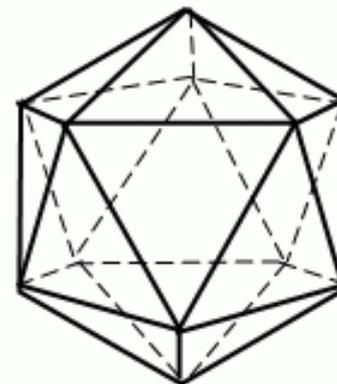


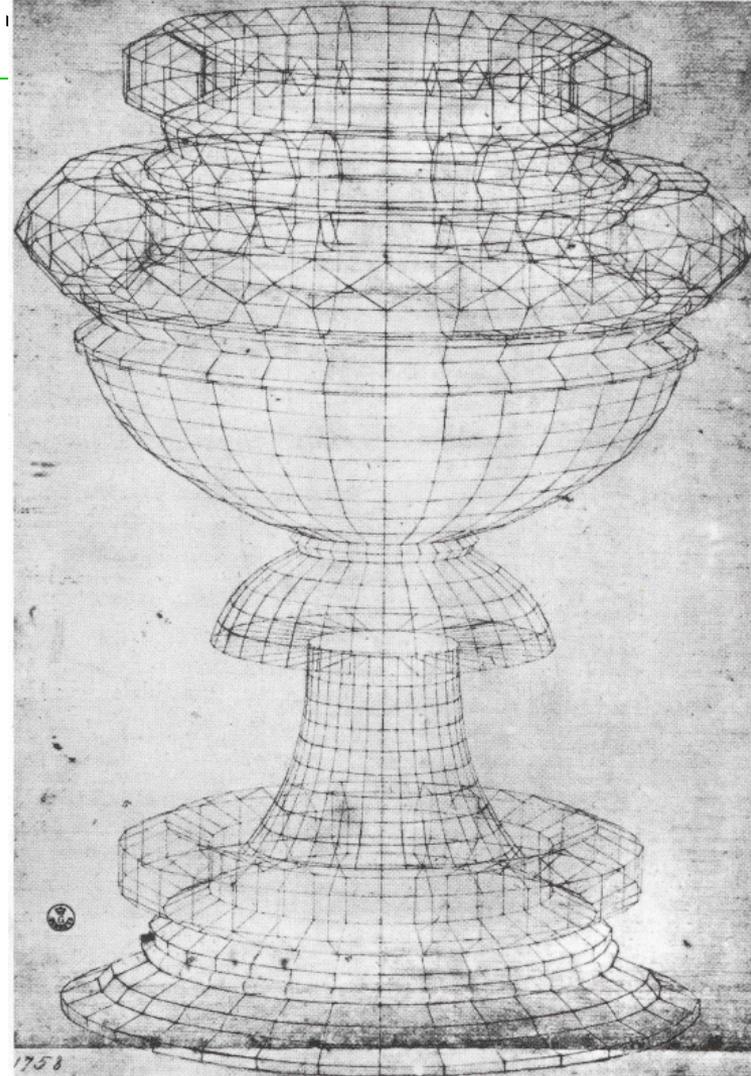
Fig. 103

Malhas Poligonais - Exemplos



Malhas Poligonais - Exemplos

<http://www.siggraph.org/education/materials/HyperGraph/modeling/polymesh/polymesh.htm>



Malhas Poligonais - Exemplos



Estruturas de Dados

- ● ● ●
- Problema

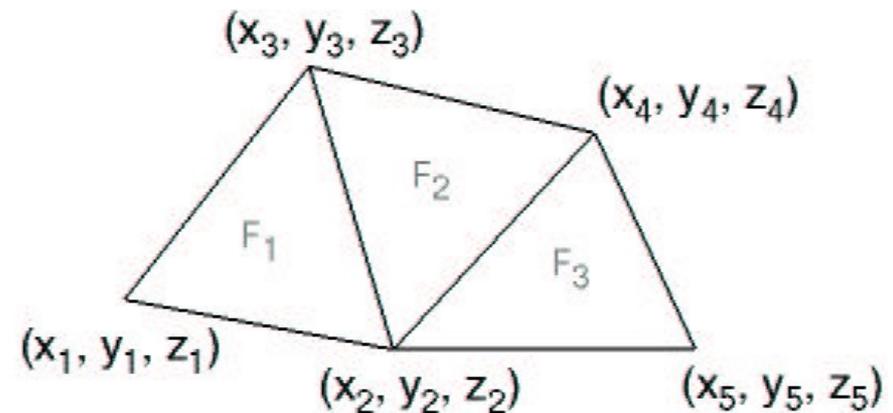
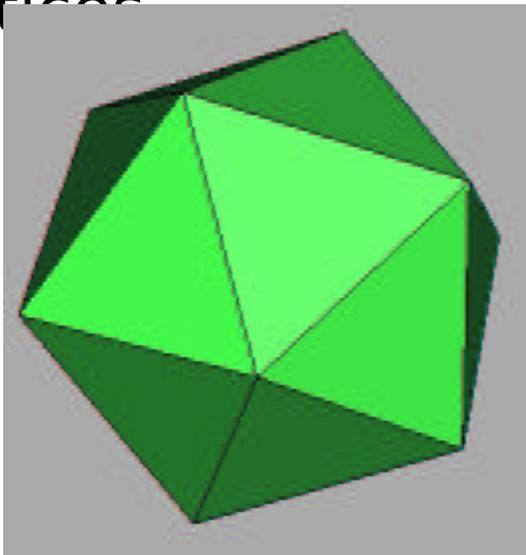
- como armazenar a descrição de um objeto em termos das faces que descrevem sua superfície?
- Diversas soluções possíveis...

- Sugestões??

Estruturas de Dados

- Solução simples

- Tabela de faces, cada face informa as coordenadas dos seus vértices



FACE TABLE			
F1	(x ₁ , y ₁ , z ₁)	(x ₂ , y ₂ , z ₂)	(x ₃ , y ₃ , z ₃)
F2	(x ₂ , y ₂ , z ₂)	(x ₄ , y ₄ , z ₄)	(x ₃ , y ₃ , z ₃)
F3	(x ₂ , y ₂ , z ₂)	(x ₅ , y ₅ , z ₅)	(x ₄ , y ₄ , z ₄)

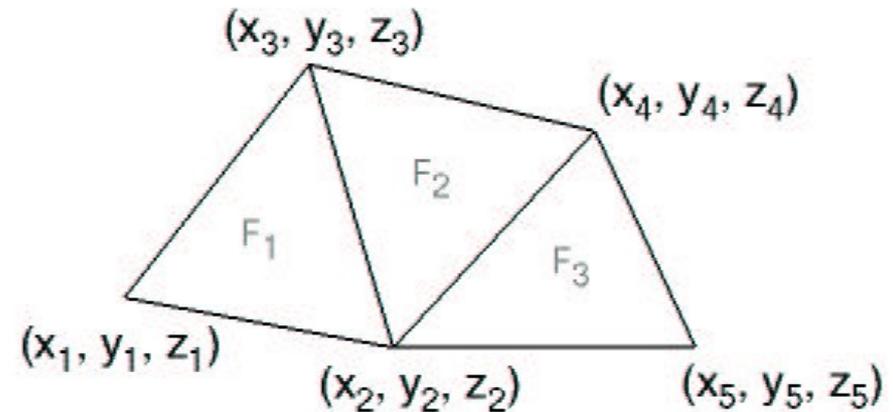
Estruturas de Dados

■ Problemas:

- vértices redundantes, ausência de informação topológica

■ Alternativa

- tabelas de faces e de vértices
- cada face lista referência aos seus vértices
- resolve os problemas?



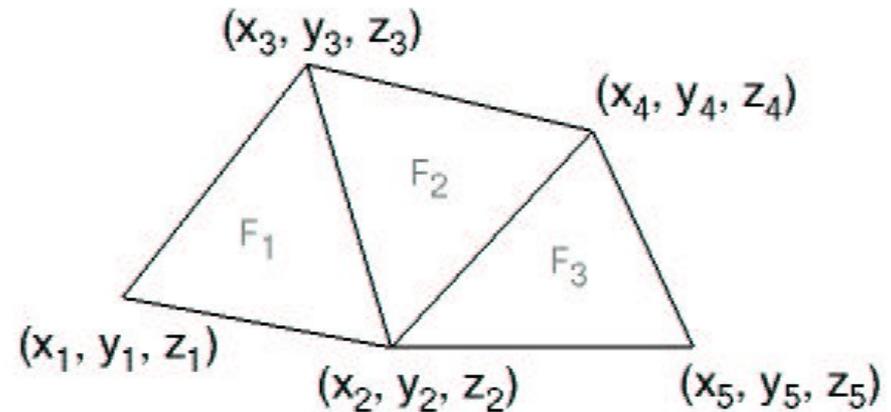
VERTEX TABLE			
V_1	X_1	Y_1	Z_1
V_2	X_2	Y_2	Z_2
V_3	X_3	Y_3	Z_3
V_4	X_4	Y_4	Z_4
V_5	X_5	Y_5	Z_5

FACE TABLE			
F_1	V_1	V_2	V_3
F_2	V_2	V_4	V_3
F_3	V_2	V_5	V_4

Estruturas de Dados

■ Outra alternativa

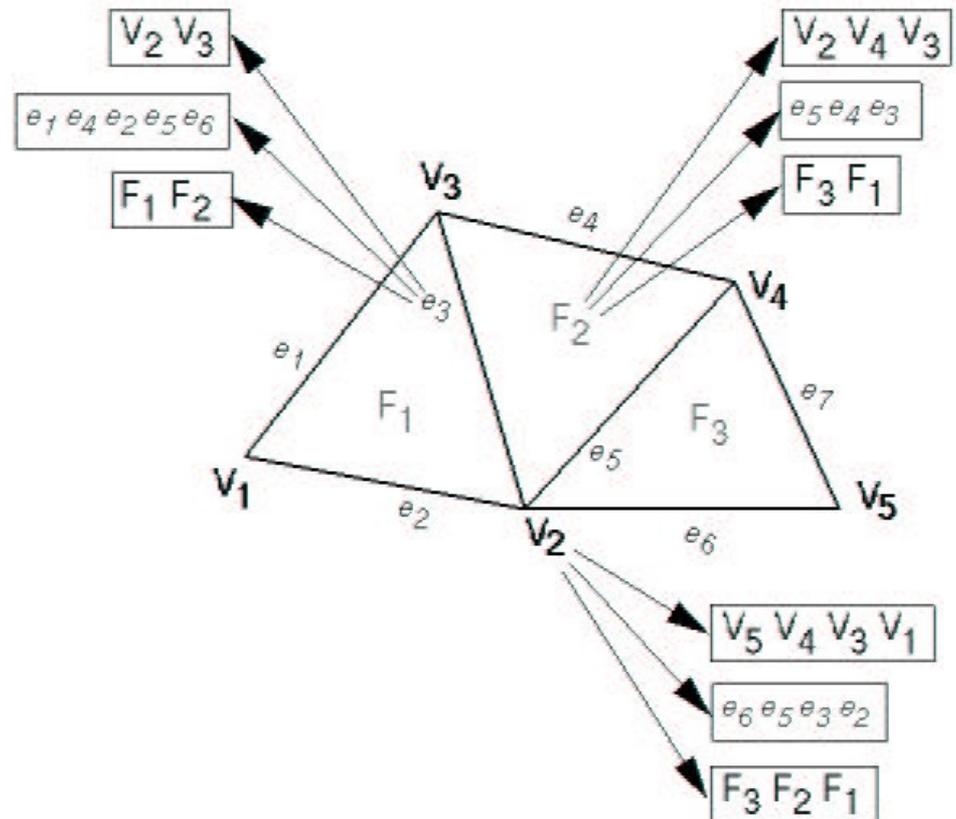
- tabelas de faces, de aresta e de vértices
- cada face lista referências às arestas que a compõem
- cada aresta lista referências aos vértices
- Todos os 'elementos topológicos' (faces, arestas e vértices) armazenados explicitamente...
- Informação topológica (adjacência) implícita



VERTEX TABLE			
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

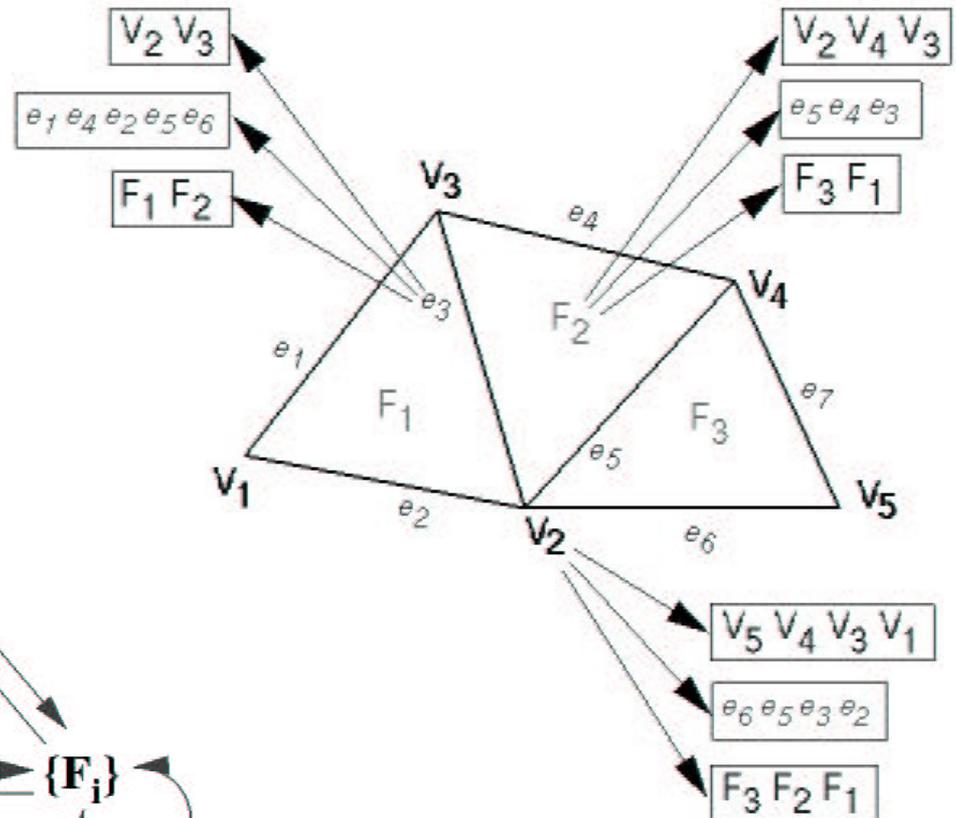
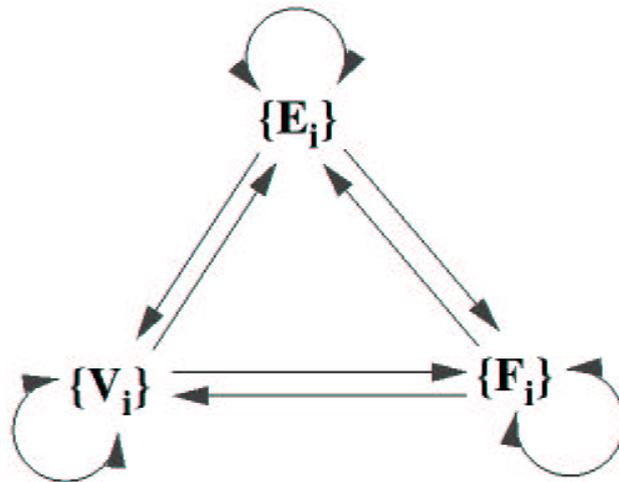
Estruturas de Dados

- Listas de adjacências
 - Armazena explicitamente todas as adjacências entre vértices, faces e arestas
 - Toda a informação topológica explícita
 - Custo extra de armazenagem



Estruturas de Dados

- Listas parciais de adjacências
 - Possível armazenar algumas relações de adjacência, e inferir outras?



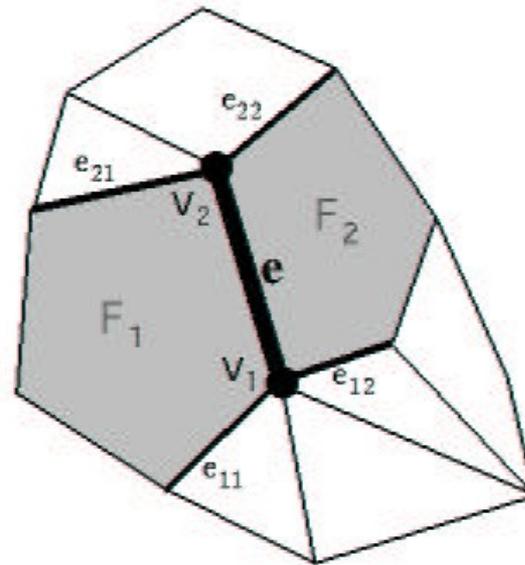
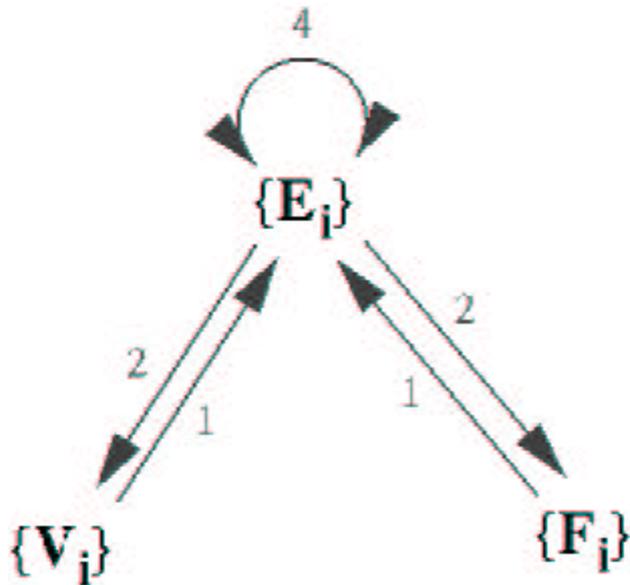
Estruturas de Dados

- Winged-edge

- Associa informações de adjacência às arestas
 - Todas as adjacências entre elementos topológicos recuperadas em tempo $O(1)$
 - Custo extra de armazenagem pequeno (registros de tamanho fixo)
 - Conseguir representar polígonos arbitrários

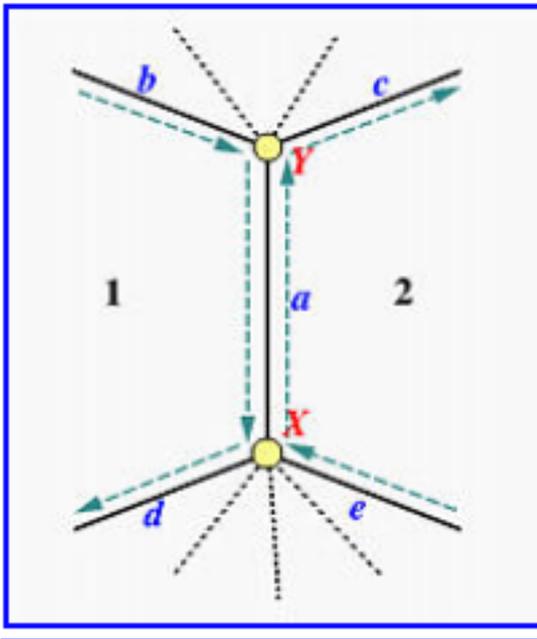
Estruturas de Dados

- Winged-edge



Estruturas de Dados

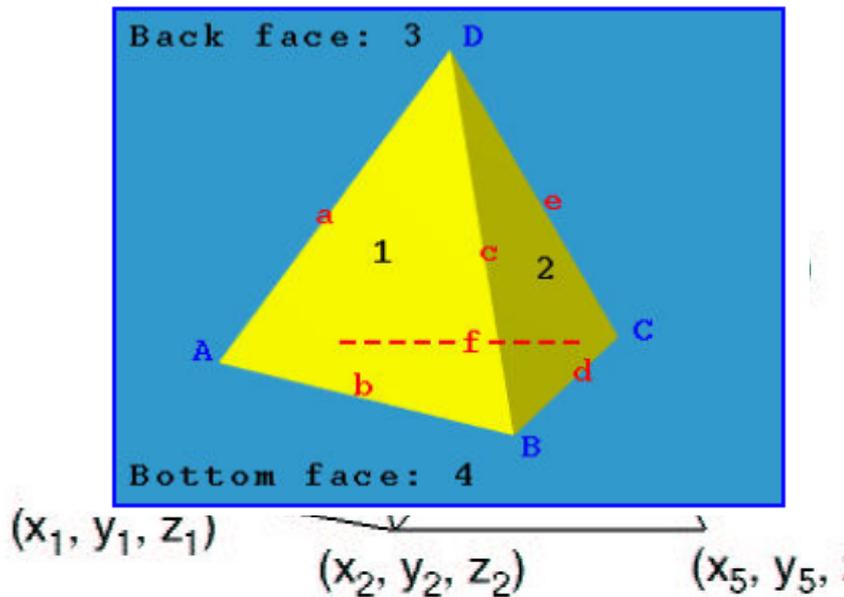
- Winged-edge - exemplo



<i>Edge</i>	<i>Vertices</i>		<i>Faces</i>		<i>Left Traverse</i>		<i>Right Traverse</i>	
<i>Name</i>	<i>Start</i>	<i>End</i>	<i>Left</i>	<i>Right</i>	<i>Pred</i>	<i>Succ</i>	<i>Pred</i>	<i>Succ</i>
<i>a</i>	<i>X</i>	<i>Y</i>	<i>1</i>	<i>2</i>	<i>b</i>	<i>d</i>	<i>e</i>	<i>c</i>

Estruturas de Dados

Winged-edge - fazer



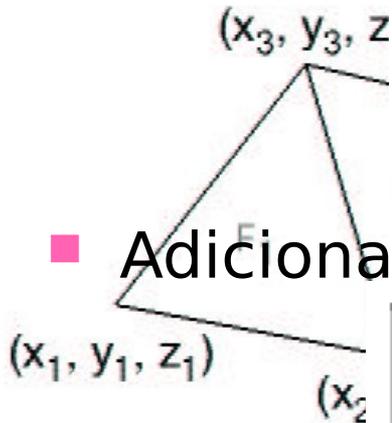
Resposta parcial (completar):

Edge	Vertices		Faces		Left Traverse		Right Traverse		
	Name	Start	End	Left	Right	Pred	Succ	Pred	Succ
a		A	D						
b		A	B						
c		B	D						
d		B	C						
e		C	D						
f		A	C						

Estruturas de Dados

- Winged-edge - resposta

Edge	Vertices		Faces		Left Traverse		Right Traverse	
	Name	Start	End	Left	Right	Pred	Succ	Pred
a	A	D	3	1	e	f	b	c
b	A	B	1	4	c	a	f	d
c	B	D	1	2	a	b	d	e
d	B	C	2	4	e	c	b	f
e	C	D	2	3	c	d	f	a
f	A	C	4	3	d	b	a	e



- Adicionalmente:

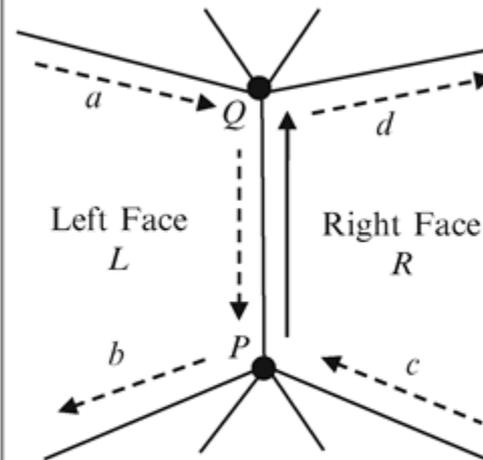
Vertex Name	Incident Edge
A	a
B	b
C	d
D	e

Face Name	Incident Edge
1	a
2	c
3	a
4	b

Winged-edge

```
struct W_edge
{
    Vertex *start, *end;
    Face   *left, *right;
    W_edge *left_prev, *left_next;
}; W_edge *right_prev, *right_next;
struct Vertex
{
    float x, y, z;
    W_edge *edge;
};
struct Face
{
    W_edge *edge;
```

(x_1, y_1, z_1)



Winged-edge - exercícios

fonte: <http://what-when-how.com/advanced-methods-in-computer-graphics/mesh-processing-advanced-methods-in-computer-graphics-part-2/>

Desenvolver o pseudo-código de uma função, usando a estrutura dada, para determinar todas as arestas incidentes a um vértice.

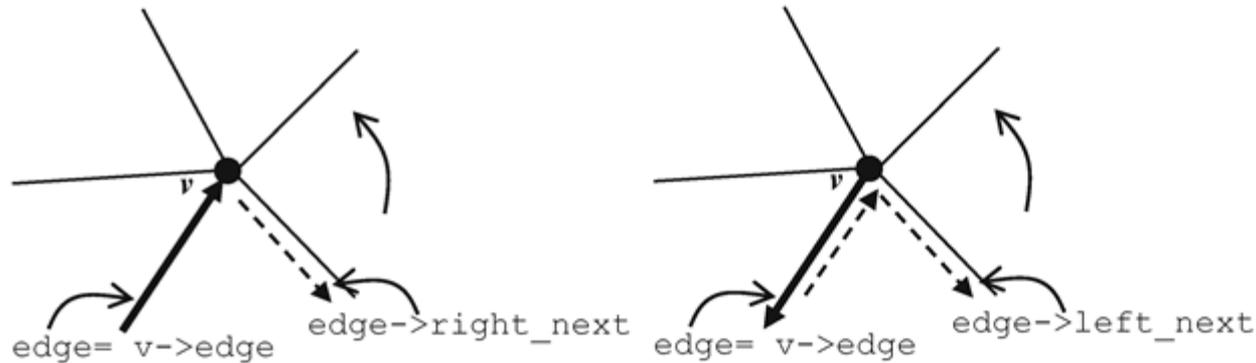
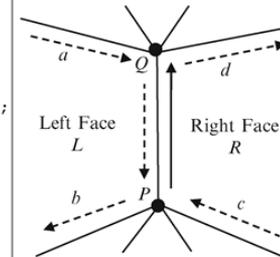


Fig. 8.11 Computation of all edges incident at a vertex. Both directions of an edge should be considered in algorithms using the winged-edge data structure

Winged-edge

Resposta:

```
struct W_edge
{
  Vertex *start, *end;
  Face *left, *right;
  W_edge *left_prev, *left_next;
};
W_edge *right_prev, *right_next;
struct Vertex
{
  float x, y, z;
  W_edge *edge;
};
struct Face
{
  W_edge *edge;
};
```



```
1. Input: v //A vertex
2. W_edge *e0 = v->edge; //Initial edge
3. W_edge *edge = e0;
4. do
5. {
6.     if(edge->end == v) edge = edge -> right_next;
7.     else edge = edge -> left_next;
8.     output(edge);
9. } while (edge != e0);
```

(x_1, y_1, z)

Winged-edge

Desenvolver o pseudo-código de uma função, usando a estrutura dada, para determinar todas as arestas que formam uma face

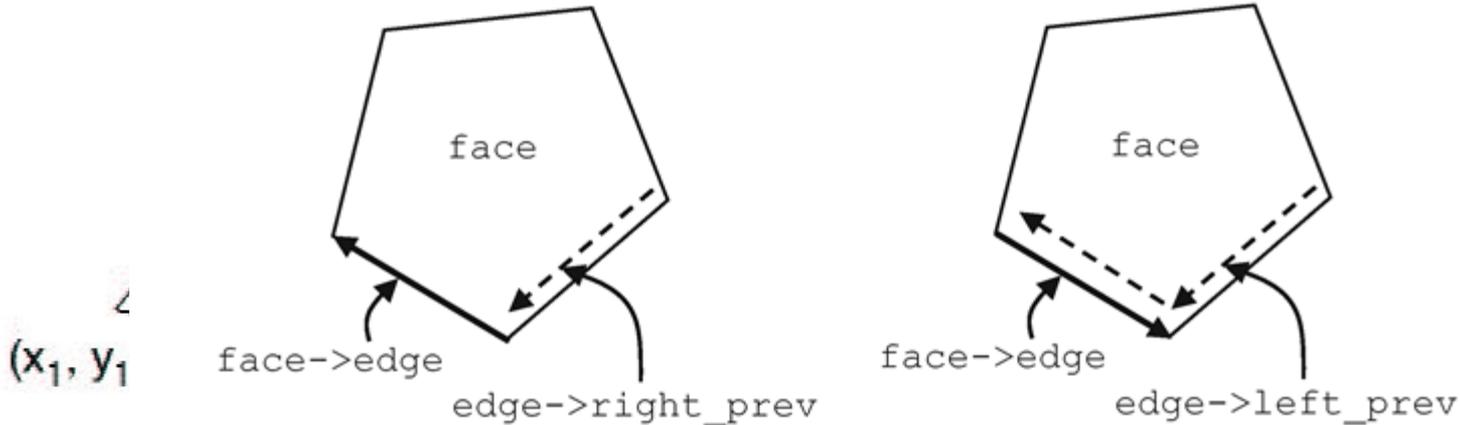


Fig. 8.12 Computation of edges around a polygonal face

Winged-edge

Resposta:

```
1. Input: face
2. W_edge *e0 = face->edge;           //Initial edge
3. W_edge *edge = e0;
4. do
5. {
6.     if(edge->right == face) edge = edge->right_prev;
7.     else edge = edge->left_prev;
8.     output(edge);
9. } while (edge != e0);
```

(x_1, y_1, z_1)

(x_2, y_2, z_2)

(x_5, y_5, z_5)

Estruturas de Dados

- Observação
 - Em muitas E.D. inconsistências podem ocorrer se o processo de geração da malha não for cuidadoso para evitá-las
 - a malha pode descrever objetos não 'factíveis'
 - Ex. vértices e arestas isolados, polígonos não fechados, polígonos isolados, etc.
- Certas estruturas, como a *winged-edge*, não admitem tais ocorrências: modelo descreve objeto consistente

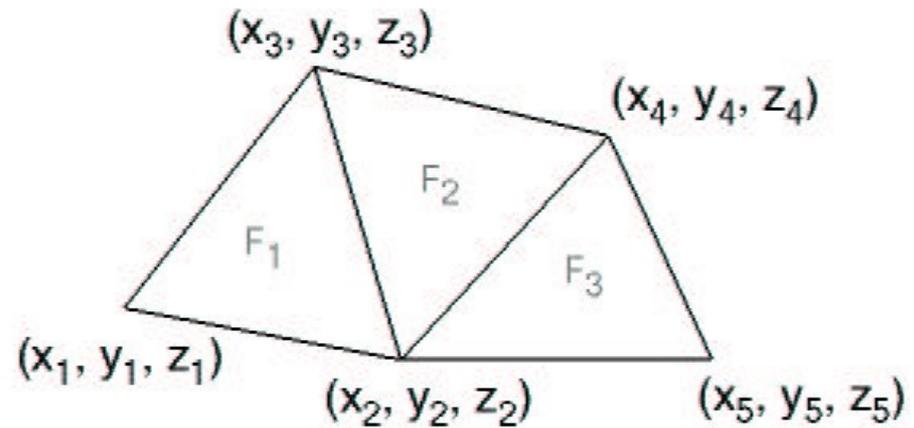
Malhas de Triângulos

■ Propriedades

- Cada face tem exatamente três vértices
- Cada vértice compartilha um número arbitrário de faces

■ Estrutura de Adjacência simples

- Faces armazenam referências aos vértices e faces vizinhas
- Maioria das relações de adjacência recuperada em tempo constante



Malhas de Triângulos

- Um problema bastante atual é o da simplificação de malhas (*decimação*)
 - Redução do número de polígonos/triângulos necessários para descrever um modelo
 - Veja:
<http://www.crd.ge.com/~lorensen/decimate/decimate.html>
 - Porquê?
 - *Rendering* mais rápido
 - Menor custo de armazenagem
 - Manipulação mais simples

Informações Geométricas

- As coordenadas dos vértices contém a informação geométrica necessária para o *rendering* do modelo
 - A partir das coord's dos vértices, pode-se computar a inclinação das arestas, o retângulo envoltório (*bounding box*) de cada face, a equação do plano que contém cada face, etc.
 - Informações necessárias para algoritmos de recorte, remoção de superfícies ocultas e de *rendering*...

Informações Geométricas

- Equação do plano:

$$Ax + By + Cz + D = 0$$

em que (x,y,z) é um ponto qqr no plano, A, B, C e D são ctes. que descrevem as propriedades espaciais do plano.

Como obter a equação do plano, dados 3 pontos que estão no plano?

v. Hearn & Baker, Seção 10.1

Informações Geométricas

- Equação do plano
- Coefs. A, B, C e D podem ser armazenados na tabela de faces
- Orientação do plano no espaço dada pela sua normal: $\mathbf{N} = (A, B, C)$
- Importante distinguir os 2 lados: 'dentro' e 'fora': especificar os vértices sempre na direção anti-horária de alguém observando o plano do lado 'de fora'

Bibliografia

- Seção 10.1 – Hearn & Baker Computer Graphics in C
- Lecture Notes by Thomas Funkhouser at Princeton University:
<http://www.cs.princeton.edu/courses/archive/fall02/cs526/lectures/meshes.pdf>