

Geometria Computacional

Professor:

Anselmo Montenegro
www.ic.uff.br/~anselmo

Conteúdo (aula 7):

- Noções de estruturas de dados topológicas

Roteiro

- Introdução
- Representação por grafos: grafo de incidências e dual
- Codificação de representação por grafos: lista de faces, vértices e arestas
- Winged-Edge
- Half-Edge

Estruturas de dados topológicas: introdução

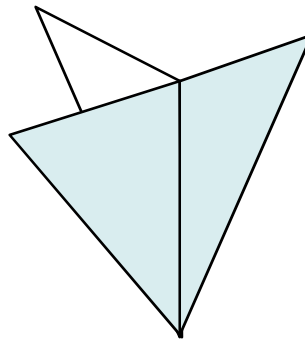
- As curvas poligonais desempenham um papel importante na representação de curvas planas.
- No caso de superfícies este papel é representado pelas *superfícies poliédricas*.
- As superfícies poliédricas se baseiam no conceito de *triangulação*.

Estruturas de dados topológicas: introdução

- Três pontos p_0 , p_1 e p_2 formam um triângulo no R^3 se os vetores $p_1 - p_0$, $p_1 - p_2$ forem *linearmente independentes*.
- Uma *triangulação 2D* no R^3 é uma coleção $\mathcal{T} = \{T_i\}$ de triângulos tal que para dois triângulos distintos T_i e T_j em \mathcal{T} , com $T_i \cap T_j \neq \emptyset$ temos:
 - $T_i \cap T_j$ é um vértice em comum ou,
 - $T_i \cap T_j$ é uma aresta em comum.

Estruturas de dados topológicas: introdução

- Uma **superfície poliédrica** é uma triangulação 2D do espaço que representa uma superfície.
- Como temos mais graus de liberdade ao posicionar os triângulos no espaço devemos evitar o seguinte caso:



- Para isso, impomos a restrição de que cada aresta seja compartilhada por apenas 2 triângulos.

Estruturas de dados topológicas: introdução

- Faces triangulares apresentam as seguintes vantagens:
 - Planaridade.
 - Sistema de coordenadas.
 - Extensibilidade.

Estruturas de dados topológicas: introdução

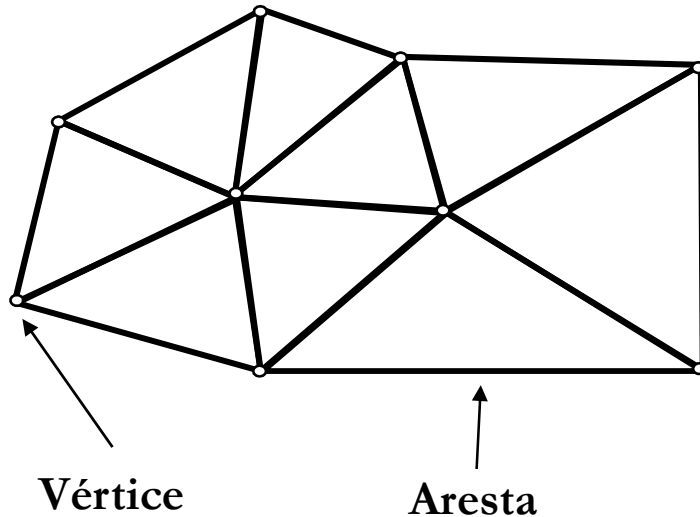
- Problema:
 - Como codificar a *estrutura geométrica e topológica* (sistema de vizinhanças) da superfície poliédrica?
- A codificação está diretamente associada a *estrutura de dados* associada a triangulação da superfície.

Estruturas de dados topológicas: representação por grafos

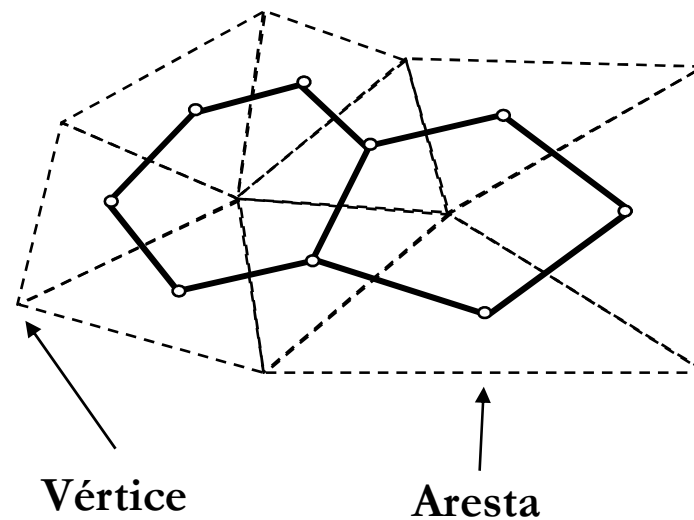
- Uma superfície poliédrica pode ser codificada através de *grafos*.
- Temos dois grafos associados a uma superfície poliedral:
 - *Grafo de vértices*
 - Induzido pelos vértices e arestas da superfície.
 - *Grafo dual*
 - Um vértice existe para cada face da superfície, os quais são conectados por uma aresta no grafo se as faces associadas são vizinhas.

Estruturas de dados topológicas: representação por grafos

Grafo de vértices



Grafo dual



- O problema de estruturação da superfície poliédrica se resume a *codificação dos grafos associados*.

Estruturas de dados topológicas: banco de dados geométrico

- A representação de uma superfície é vista como um *banco de dados geométrico*.
- É comum efetuar certos tipos de consulta sobre propriedades geométricas e topológicas da superfície:
 - Achar todas as arestas que incidem em um vértice.
 - Achar todos os polígonos que compartilham uma aresta ou um vértice.
 - Achar as arestas que delimitam um polígono.
 - Visualizar a superfície.

Objetos gráficos espaciais: codificação de superfícies poliédricas

- A escolha da codificação está intimamente ligada ao conjunto de operações que se deseja realizar.
- Veremos 3 tipos de codificação:
 - Codificação explícita.
 - Codificação por lista de vértices.
 - Codificação por lista de arestas.

Fecho convexo: estruturas de dados para representação de poliedros

- Codifica explicitamente os polígonos da superfície fornecendo uma lista de vértices com suas coordenadas.

Codificação explícita

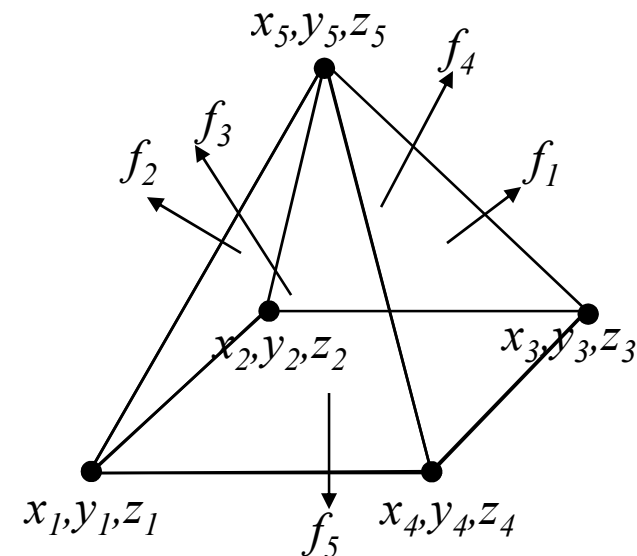
$$f_1 = ((x_3, y_3, z_3), (x_5, y_5, z_5), (x_4, y_4, z_4))$$

$$f_2 = ((x_1, y_1, z_1), (x_5, y_5, z_5), (x_2, y_2, z_2))$$

$$f_3 = ((x_1, y_1, z_1), (x_4, y_4, z_4), (x_5, y_5, z_5))$$

$$f_4 = ((x_3, y_3, z_3), (x_2, y_2, z_2), (x_5, y_5, z_5))$$

$$f_5 = ((x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4))$$



Estruturas de dados topológicas: representação por lista de faces

- Vantagens: Extremamente simples.
- Desvantagens - redundância :
 - Ocupa *espaço de armazenamento desnecessário*.
 - Operações geométricas *podem introduzir erros numéricos* independentes nas coordenadas dos vértices.
 - *Ineficiência* (cada aresta é desenhada duas vezes na visualização).

Estruturas de dados topológicas: representação por lista de faces

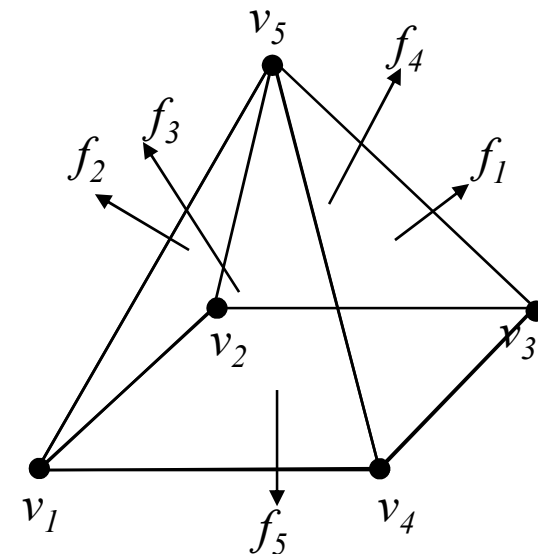
- Para solucionar os problemas encontrados na codificação explícita devemos eliminar os seguintes problemas:
 - *Evitar a replicação de vértices.*
 - Codificar as *informações de adjacência.*

Estruturas de dados topológicas: representação por lista de vértices

- Criamos uma *lista de vértices* e cada polígono da superfície é definido por referência aos vértices desta lista.

Lista de vértices
$v_1 = (x_1, y_1, z_1)$
$v_2 = (x_2, y_2, z_2)$
$v_3 = (x_3, y_3, z_3)$
$v_4 = (x_4, y_4, z_4)$
$v_5 = (x_5, y_5, z_5)$

Lista de faces
$f_1 = (v_3, v_5, v_4)$
$f_2 = (v_1, v_5, v_2)$
$f_3 = (v_1, v_4, v_5)$
$f_4 = (v_3, v_2, v_5)$
$f_5 = (v_1, v_2, v_3, v_4)$



Estruturas de dados topológicas: representação por lista de vértices

- Vantagens:
 - Proporciona *maior economia de espaço*.
 - Ao alterar as coordenadas de um vértice, todos os polígonos nele incidentes são alterados automaticamente.
- Ainda alguns problemas:
 - É difícil determinar os polígonos que compartilham uma aresta.
 - Arestas compartilhadas são desenhadas duas vezes.

Estruturas de dados topológicas: representação por lista de arestas

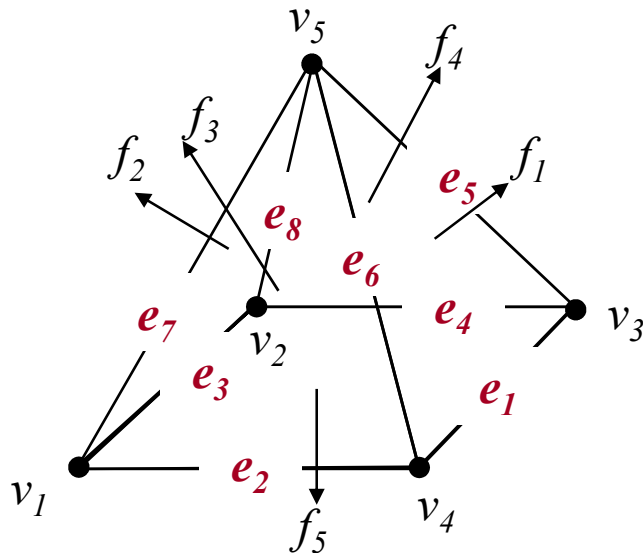
- Acrescentamos uma *lista de arestas* definida por pares de referências à lista de vértices.
- A *lista de faces* é definida por referências às arestas que as definem, descritas na lista de arestas.

Estruturas de dados topológicas: representação por lista de arestas

Lista de vértices
$v_1 = (x_1, y_1, z_1)$
$v_2 = (x_2, y_2, z_2)$
$v_3 = (x_3, y_3, z_3)$
$v_4 = (x_4, y_4, z_4)$
$v_5 = (x_5, y_5, z_5)$

Lista de arestas
$e_1 = v_3, v_4$
$e_2 = v_4, v_1$
$e_3 = v_1, v_2$
$e_4 = v_2, v_3$
$e_5 = v_3, v_5$
$e_6 = v_5, v_4$
$e_7 = v_5, v_1$
$e_8 = v_5, v_2$

Lista de faces
$f_1 = e_1, e_5, e_6$
$f_2 = e_3, e_7, e_8$
$f_3 = e_2, e_6, e_7$
$f_4 = e_4, e_8, e_5$
$f_5 = e_1, e_2, e_3, e_4$

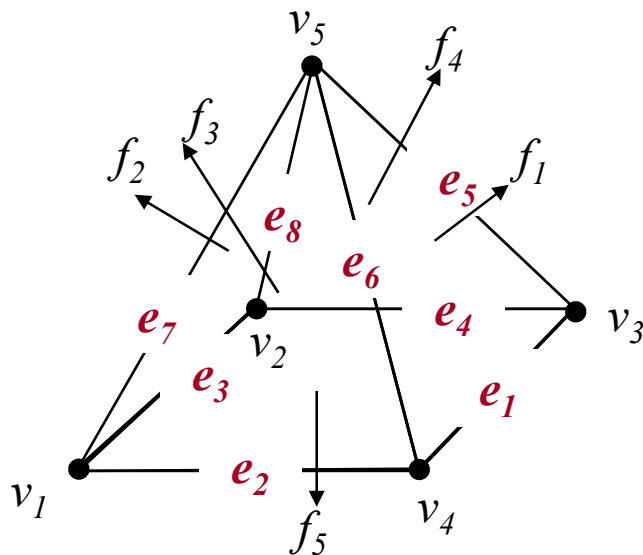


Estruturas de dados topológicas: representação por lista de arestas

- Propriedades
 - Acesso a todas as arestas sem precisar percorrer as fronteiras dos polígonos.
 - As arestas que incidem em um vértice podem ser obtidas através de uma combinação de algoritmos geométricos e de busca.

Estruturas de dados topológicas: representação por listas de arestas

- Podemos acrescentar na lista de arestas informações sobre as faces adjacentes a uma aresta.



Lista de arestas

$$e_1 = v_3, v_4, f_1, f_5$$

$$e_2 = v_4, v_1, f_3, f_5$$

$$e_3 = v_1, v_2, f_2, f_5$$

$$e_4 = v_2, v_3, f_4, f_5$$

$$e_5 = v_3, v_5, f_1, f_4$$

$$e_6 = v_5, v_4, f_1, f_3$$

$$e_7 = v_5, v_1, f_2, f_3$$

$$e_8 = v_5, v_2, f_2, f_4$$

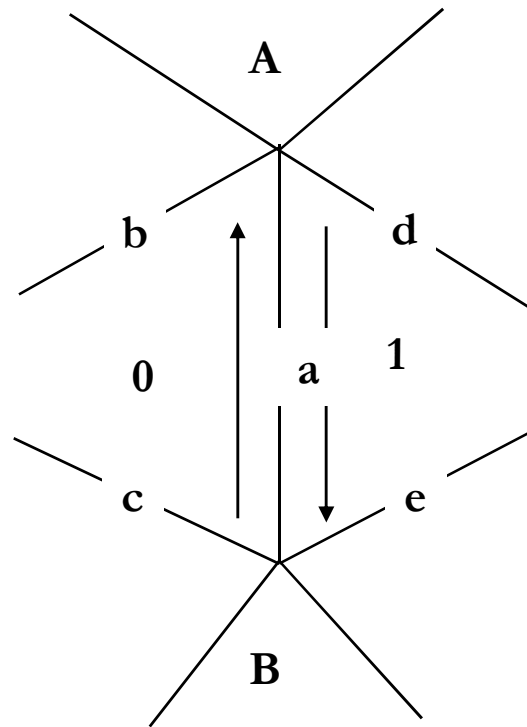
Estruturas de dados topológicas: estruturas de dados clássicas

- As codificações descritas anteriormente ainda possuem muitas restrições quanto à representação da topologia das faces e da geometria do objeto gráfico.
- Codificações mais completas são dadas pelas estruturas topológicas clássicas como, por exemplo:
 - *Winged-edge*
 - *Half-edge*
 - *Radial-edge*

Estruturas de dados topológicas: Winged-edge

- A aresta é o elemento fundamental desta estrutura de dados.
- Juntamente com cada aresta são armazenadas as faces (polígonos) à direita e à esquerda.
- São também armazenadas para cada aresta as arestas sucessoras e predecessoras na ordem de percorrimento de cada uma de suas faces.

Estruturas de dados topológicas: Winged-edge

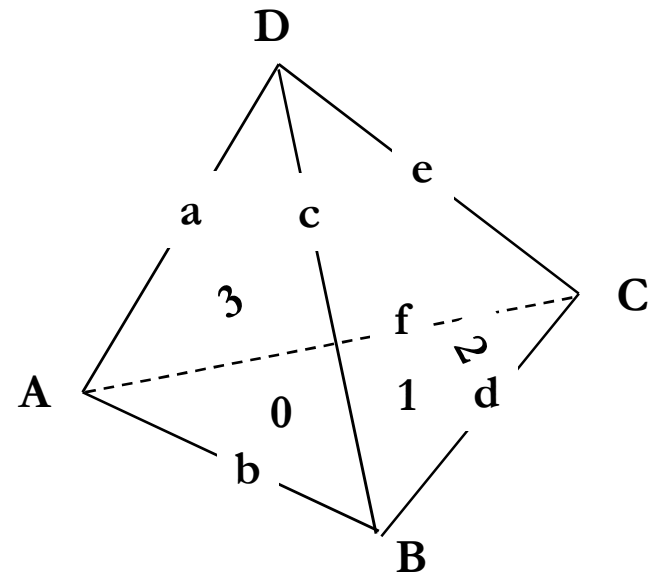


aresta	vértice1	vértice2	face esq	face dir	pred esq	succ esq	pred dir	succ dir
a	B	A	0	1	c	b	d	e

Estruturas de dados topológicas: Winged-edge

Vértice	aresta
A	a
B	d
C	d
D	e

Vértice	aresta
0	a
1	c
2	d
3	a



aresta	vértice1	vértice2	face esq	face dir	pred esq	succ esq	pred dir	succ dir
a	A	D	3	0	f	e	c	b
b	A	B	0	2	a	c	d	f
c	B	D	0	1	b	a	e	d
d	B	C	1	2	c	e	f	b
e	C	D	1	3	d	c	a	f
f	C	A	3	2	e	a	b	d

Estruturas de dados topológicas: Winged-edge

- Obs: as duas tabelas de vértices e faces não são únicas.
- As consultas são feitas em tempo *constante*.
- Uma face pode acessar uma de suas arestas e *percorrer os ponteiros* para encontrar todas as suas arestas.

Estruturas de dados topológicas: Half-edge

- Na estrutura winged-edge, para cada aresta, mantemos referências para as duas faces adjacentes
- Isto porque cada aresta é de fato compartilhada por duas faces

Estruturas de dados topológicas: Half-edge

- A ideia da estrutura Half-Edge é considerar um aresta como composta por duas semi-arestas (as Half-edges)
- Cada half-edge é adjacente somente a uma face

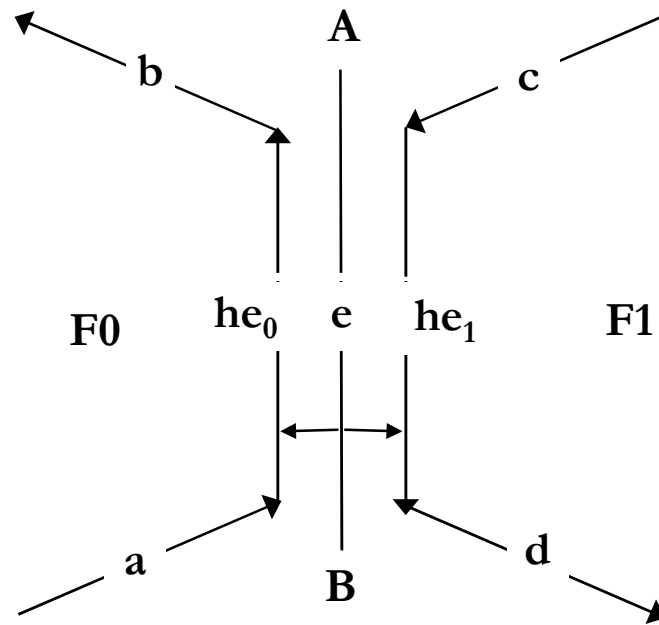
Estruturas de dados topológicas: Half-edge

- Com efeito, guardamos apenas uma referência para uma face em cada half-edge
- Além disso, mantemos apenas referências para a half-edge anterior e posterior na sequência de half-edges que compõem a face

Estruturas de dados topológicas: Half-edge

- Duas half-edges são consideradas gêmeas se correspondem a semi-arestas de uma mesma aresta
- Sejam he_0 e he_1 as half-edges à esquerda e à direita de uma aresta e
- Dizemos então que $he_0 = twin(he_1)$ e $he_1 = twin(he_0)$

Estruturas de dados topológicas: Half-edge



Estruturas de dados topológicas: Referências

- Fundamentos:
J. Gomes e L. Velho, *Fundamentos da Computação Gráfica* - IMPA
- Winged-edge:
P. C. P. Carvalho e L. H. de Figueiredo,
Introdução à Geometria Computational, 18^o Colóquio Brasileiro de Matemática, IMPA, 1991.
- Half-edge:
M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf,
[*Computational Geometry: Algorithms and Applications*](#), Springer-Verlag, 1997.