

Métodos de Modelagem Representação de Sólidos

Mortenson 2006, Cap 11

Conci, Cap4

Foley 1996, Cap 12



Métodos de Modelagem (Rep.)

(Gomes & Velho 1990: cap 7)

◆ Representação Construtiva

- Boudaring Representation
- Construtive Solid Geometry

◆ Representação por Decomposição

- Uniforme
- Não Uniforme



Boundary Representation (B-rep)

Representação por Superfícies Limítrofes
Representação por Bordos
Representação por Fonteira

Mortenson 2006: Cap11.2 e 11.6

Foley 1996 12.5



Conteúdo

- ◆ Definição de B-rep
 - Superfícies Orientáveis
- ◆ Tipos de B-rep
- ◆ B-rep Sólido Poliédrico
- ◆ B-rep Generalizado
- ◆ Elementos e Características
- ◆ Validação
- ◆ Esquemas de Representação para B-rep
 - Triângulos
 - Operações
 - Estruturas de Dados



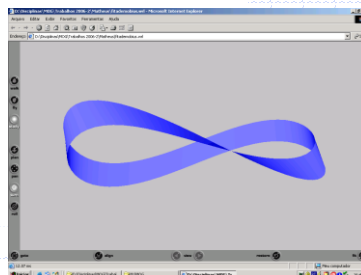
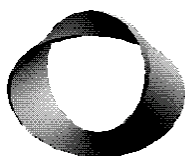
Representação por Bordos (Mortenson 2006)

- ◆ Descreve completamente um objeto sólido como uma coleção organizada de suas superfícies limitantes (:377)
- ◆ Uma superfície limitante separa pontos que são internos e externos ao sólido (:379)
- ◆ As superfícies do sólido deve atender às seguintes condições (:377):
 - Fechadas, Orientáveis, Não auto-intersectantes, todas conectadas, e todas limitantes do objeto.



Superfícies Não-orientáveis

Visualização Forçada da
Fita de Möbius



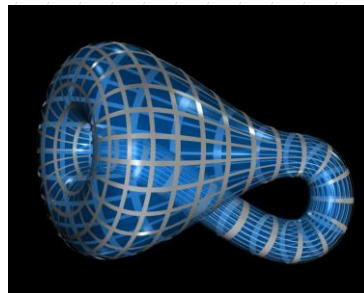
Superfícies Não-orientáveis

Fotografia de **Fita de Möbius** (By David Benbennick - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=50359>)

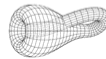


Superfícies Não-orientáveis

Visualização Forçada da Garrafa de Klein



Visualização da evolução de uma Figura Klein



By Rolcott - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=44077162>



Tipos de B-reps

- ◆ B-reps generalizados (sup. suaves, arestas curvas)
 - Patches
 - Flexíveis
 - Exatos
- ◆ B-reps poliédricos (sup. planares, arestas retas)
 - Simplificados
 - Eficientes
 - Aproximados

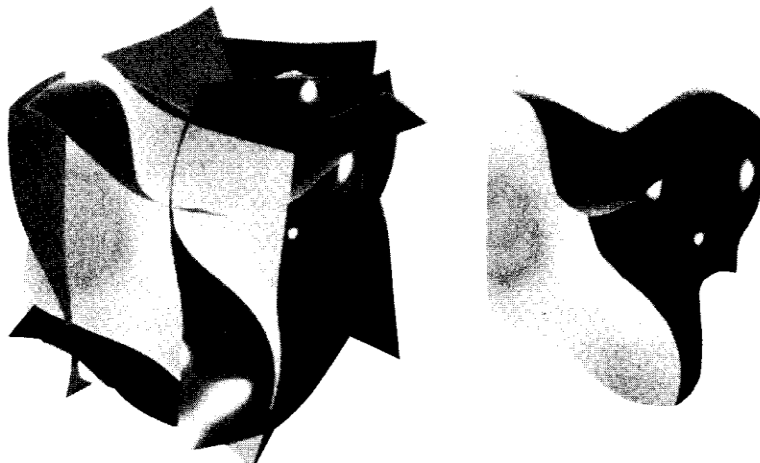


B-reps Generalizados

- ◆ Faces podem ser “retalhos” (*patches*) de superfícies livres se o algoritmo que trata a representação conseguir trabalhar com as curvas de interseção resultantes (grau maior de complexidade)



B-reps (Foley96:548)

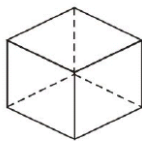


B-rep Poliédrico Foley 1996 12.5.1

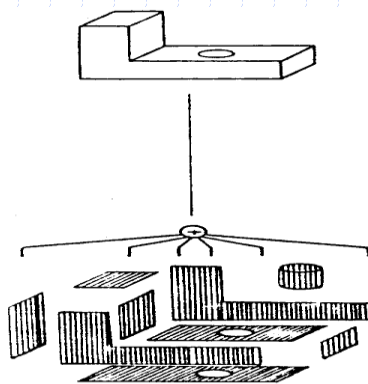
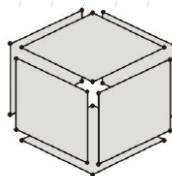
- ◆ Um B-rep simplificado representa um poliedro, ou seja:
 - O objeto é limitado por superfícies planares poligonais
 - As faces são limitadas por arestas retas
 - As arestas são limitadas por dois vértices no R^3



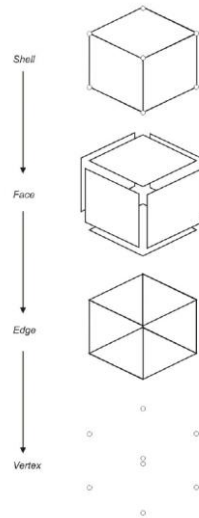
B-reps



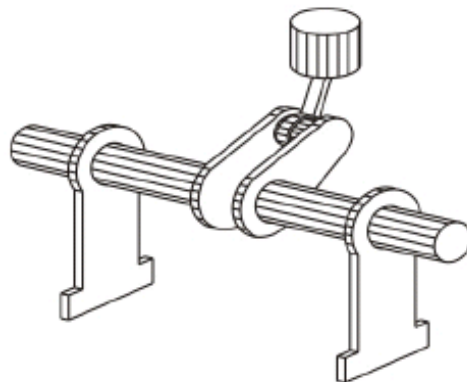
(a)



Representação Hierárquica dos Elementos Topológicos



Exemplo de B-rep Poliédrico

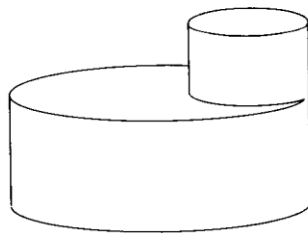


B-reps (O que é uma face?)

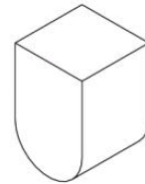
(Foley:542)

- ◆ Caso restrições não sejam usadas pode ser bastante difícil determinar o que é uma face

(ver Foley96:542)



How many faces does this object have?



B-reps (FACES e Arestas)

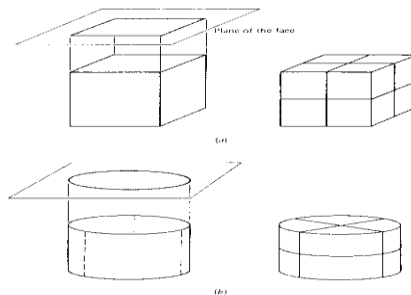


FIGURE 10.35 Faces defining the boundary of a solid.

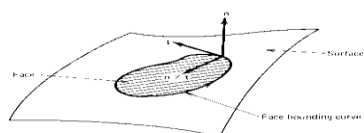


FIGURE 10.36 Face boundary convention.



Características do B-rep

◆ Limitações

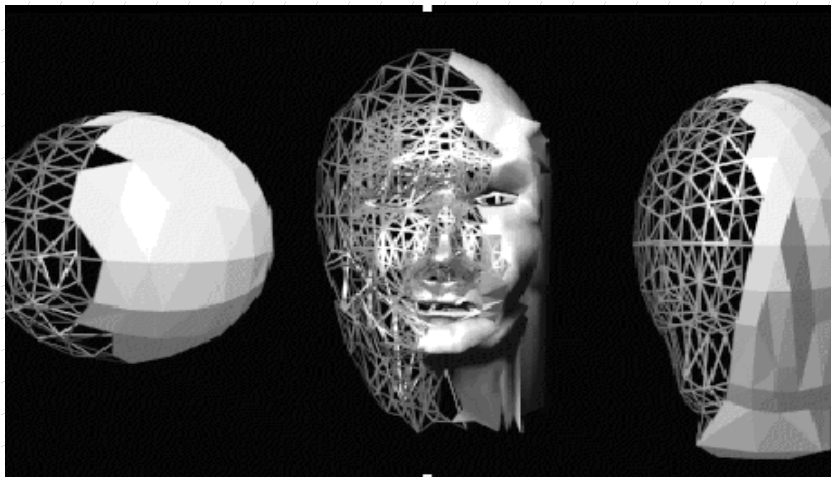
- Conceito de Superfície/Face nem sempre é claro e facilmente representável
- Não unicidade construtiva

◆ Vantagens

- Dados *Wireframe* podem ser facilmente derivados dos modelos B-rep baseados em grafos
- Pode-se obter B-rep de várias outras formas de representação e criação



B-reps Semi-renderizados



Esquemas de Representação Brep Sólidos Poliédricos



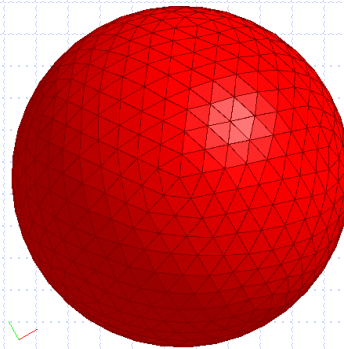
Poliedros

- ◆ Objetos Complexos podem ser divididos em vários polígonos simples
- ◆ Os polígonos são a “pele” do objeto
 - Objetos Ocos
 - Polígonos têm face frontal e traseira



Representação Linear por Partes

- ◆ Objeto parametrizado com geometria complexa pode ser aproximado por uma superfície linear por partes
- ◆ Pode-se particionar o domínio da parametrização por um conjunto de polígonos.
 - Cada vértice no domínio poligonal é levado para a superfície pela parametrização.
 - Em seguida é ligado aos vértices adjacentes mantendo as conectividades do domínio.



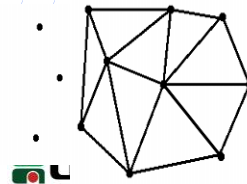
Triângulos

- ◆ Triângulos são a primeira escolha de polígonos
 - Eles são sempre planares
 - Hardware gráfico são otimizados para triângulos
 - ◆ Nvidia TITAN X:
 - 3584 núcleos NVIDIA CUDA
 - Processa a 11TFLOPS
 - Obs.: Não se apeguem muito a estes números. São similares aos número que dizem de velocidade de processadores. Eles não dizem de fato o quão rápido é o computador.
 - Ver nas versões 2024/2025.



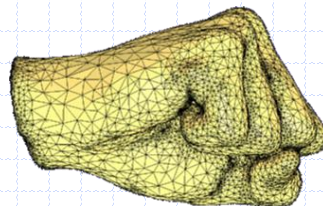
Triangularização

- ◆ Como os triângulos são obtidos?
 - Triangularizando um conjunto de pontos da superfície
- ◆ Existem várias técnicas diferentes
 - Triangularização de Delaunay tenta equalizar o ângulos internos (evita triângulos finos e longos)



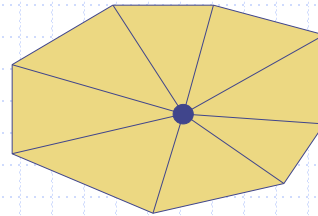
Quantos triângulos?

- ◆ Quanto maior o detalhamento geométrico, maior será a necessidade de mais triângulos
 - Higher curvature → more triangles necessary

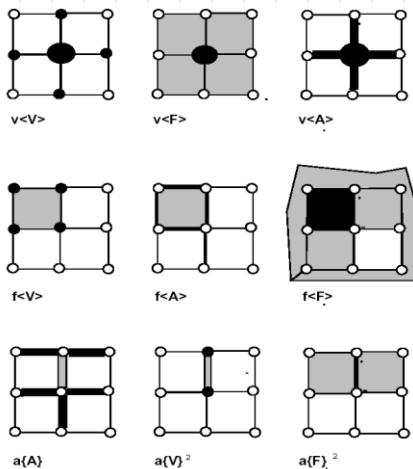


Operações Importantes sobre Malhas Poligonais

- ◆ Desenhar a malha.
- ◆ Identificar adjacências
 - Achar todas as arestas que incidem em um vértice.
 - Achar as faces que incidem numa aresta ou vértice.
 - Achar as arestas na fronteira de uma face.
 - ...



9 tipos de adjacências



Estruturas de Dados para B-reps

- ◆ **Sólidos Poliédricos (2-manifold)**
- ◆ 3-manifold
- ◆ N-manifold
- ◆ Non-manifold

LAGE, LEWINER, LOPES e VELHO.

Editor Escalável para Malhas Tetraédricas.

SIBGRAPI '2005, pp 349-...



Manifolds and Shapes

- ◆ Curvas → 1D-manifolds
- ◆ Superfícies → 2D-manifolds
- ◆ Sólidos → 3D-manifolds with boundary
- ◆ "2D-sólidos" → 2D-manifolds with boundary



Non-Manifold

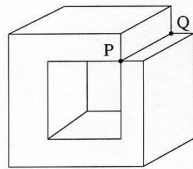


Figure 2.28 A Non-manifold Object

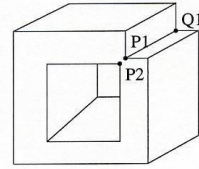
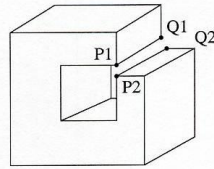


Figure 2.29 Two Possible Topologies



Estrutura de Dados para B-rep

◆ “A melhor estrutura de dados para representar um B-rep é um grafo hierárquico” (Mortenson 2006: 380)

◆ Lembrete “árvore e listas são casos simplificados de grafos”



B-rep Sólidos 2-manifold

- ◆ Codificação Explícita
- ◆ Lista de Vértices
- ◆ Lista de Arestas
- ◆ Winged-Edge de Weiler, 1988 (Foley 96: 2.5.2)
- ◆ Half-Edge de Mantyla, 1988
- ◆ Quad-Edge de Guibas e Stolfi, 1985
- ◆ Handle-Edge
- ◆ Corner Table
- ◆ Direct-Edges



Codificação explícita

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

Codificação explícita
$f_1 = ((x_1, y_1, z_1), (x_5, y_5, z_5), (x_2, y_2, z_2))$
$f_2 = ((x_3, y_3, z_3), (x_2, y_2, z_2), (x_5, y_5, z_5))$
$f_3 = ((x_3, y_3, z_3), (x_4, y_4, z_4), (x_5, y_5, z_5))$
$f_4 = ((x_1, y_1, z_1), (x_4, y_4, z_4), (x_5, y_5, z_5))$
$f_5 = ((x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4))$



Codificação explícita

◆ Vantagens:

- Extremamente simples.

◆ Desvantagens:

- **Redundância** (não considera que os vértices são compartilhados):
 - ◆ Ocupa espaço de armazenamento desnecessário.
 - ◆ Operações geométricas introduzem erros numéricos independentes nas coordenadas dos vértices.
 - ◆ Ineficiência (cada aresta é desenhada duas vezes na visualização).



Propriedades desejadas em uma codificação

◆ Para solucionar os problemas encontrados na codificação explícita devemos eliminar os seguintes problemas

- Evitar a replicação de vértices.
- Codificar as informações de adjacência.



Lista de Vértices

◆ Definição

- Há uma **lista de vértices** armazenados separadamente (geometria)
- Faces **listam os vértices** que as compõem (topologia).

◆ Características

- Proporciona maior economia de memória.
- É um esquema simples e rápido.
- Achar adjacências é complicado.
- As arestas são desenhadas duas vezes.



Codificação por lista de vértices

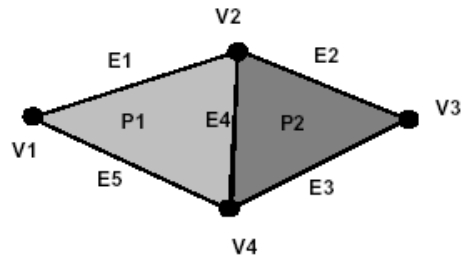
- ◆ Criamos uma **lista de vértices** e cada polígono da superfície é definido por referência aos vértices desta lista.

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



Lista de Vértices



- $V = \{V_1 = (x_1, y_1, z_1), V_2 = (x_2, y_2, z_2), V_3 = (x_3, y_3, z_3), V_4 = (x_4, y_4, z_4)\}$;
- $P_1 = \{V_1, V_2, V_4\}$;
- $P_2 = \{V_4, V_2, V_3\}$.

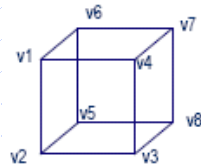


Codificação por lista de vértices

- ◆ Proporciona maior economia de espaço (que anterior)
- ◆ Ao alterar as coordenadas de um vértice, todos os polígonos nele incidentes são alterados automaticamente.
- ◆ É difícil determinar os polígonos que compartilham uma aresta.
- ◆ Arestas compartilhadas são desenhadas duas vezes.



Wireframe versus ListadeVértices



Vértices={ v1,...,v8}
Arestas={v1-v2,v2-v3,...,v1-v6}

Vértices={ v1,...,v8}
Faces={{v1,v2,v3,v4}, ..., (v2,v5,v8,v3)}

L-V = Incorpora o conceito de "face" na estrutura mas, não armazena/considera as arestas (não estão prontamente disponíveis)



Lista de Arestas

- ◆ Tem-se uma **lista de arestas** (sem repeti-las) que indicam os vértices que as compõem
- ◆ Faces apontam para a **lista de arestas** e cada aresta inclui referência para as duas faces que compartilham uma aresta (redundância).
- ◆ Facilita a determinação das duas faces incidentes na aresta.
- ◆ Arestas são prontamente desenhadas percorrendo-se a lista de arestas.



Codificação 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.

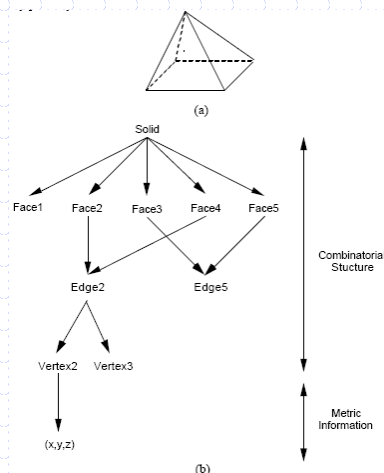
Lista de vértices
$v_1 = (x_1, y_1, z_1)$
$v_2 = (x_2, y_2, z_2)$
$v_3 = (x_3, y_3, z_3)$
$v_4 = (x_4, y_4, z_4)$
$v_5 = (x_4, y_4, z_4)$

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

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



Estrutura de Dados para B-rep Poliédrico



Codificação por lista de arestas

◆ Propriedades

- Temos 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.



Codificação por lista de arestas

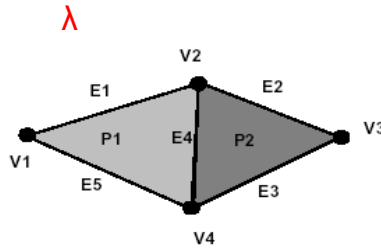
- ◆ Podemos acrescentar na lista de arestas informações sobre as faces adjacentes a uma aresta.

Lista de arestas
$e_1 = v_1, v_2, f_1, f_5$
$e_2 = v_2, v_3, f_3, f_5$
$e_3 = v_3, v_4, f_2, f_5$
$e_4 = v_4, v_1, f_4, f_5$
$e_5 = v_1, v_5, f_1, f_4$
$e_6 = v_2, v_5, f_1, f_3$
$e_7 = v_3, v_5, f_2, f_3$
$e_8 = v_4, v_5, f_2, f_4$



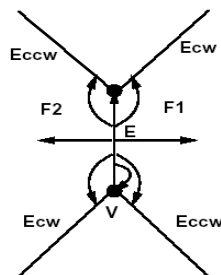
Listas de Arestas

- $V = \{V_1 = (x_1, y_1, z_1), V_2 = (x_2, y_2, z_2), V_3 = (x_3, y_3, z_3), V_4 = (x_4, y_4, z_4)\};$
- $E_1 = \{V_1, V_2, P_1, \lambda\};$
- $E_2 = \{V_2, V_3, P_2, \lambda\};$
- $E_3 = \{V_3, V_4, P_2, \lambda\};$
- $E_4 = \{V_2, V_4, P_1, P_2\};$
- $E_5 = \{V_4, V_1, P_1, \lambda\};$
- $P_1 = \{E_1, E_4, E_5\};$
- $P_2 = \{E_2, E_3, E_4\}.$



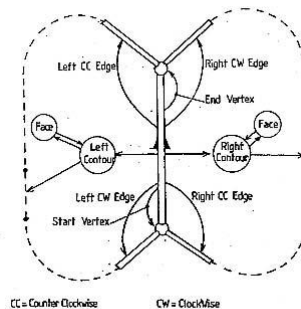
Winged-Edge Foley 1996 12.5.2

- ◆ Criada entre 1972 e 1975 por Baumgart
- ◆ Foi um marco na representação por fronteira



Winged-Edge Stroud 2006

Braid, I. C. "Designing with volumes", CANTAB Press,
Cambridge Computer Laboratory, University of
Cambridge (Ph.D. Dissertation), 1974



Winged Edge Foley 1996 12.5.2

- ◆ Armazena mais informações topológicas
- ◆ Facilita a busca de elementos relacionados
- ◆ Facilita verificação de consistência (validação topológica)
- ◆ Armazena informação na estrutura associada às arestas com número pequeno de campos (compacto) e fixo



Winged Edge Foley 1996 12.5.2

- ◆ Permite obter todos os 9 tipos de adjacência entre vértices, arestas e faces
- ◆ Permite determinar quais faces ou vértices estão adjacentes a aresta em tempo constante mas, outros tipos de adjacências podem requerer um processamento maior.
- ◆ Atualizada com o uso de operadores de Euler
- ◆ Foi concebido para faces que não têm furos, *a priori*.

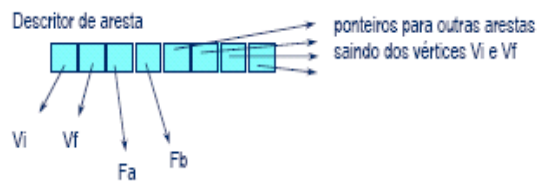
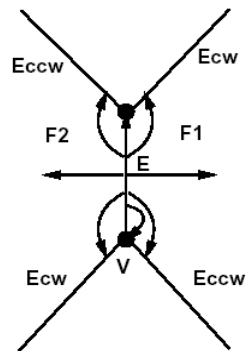


Estrutura de Dados W-E

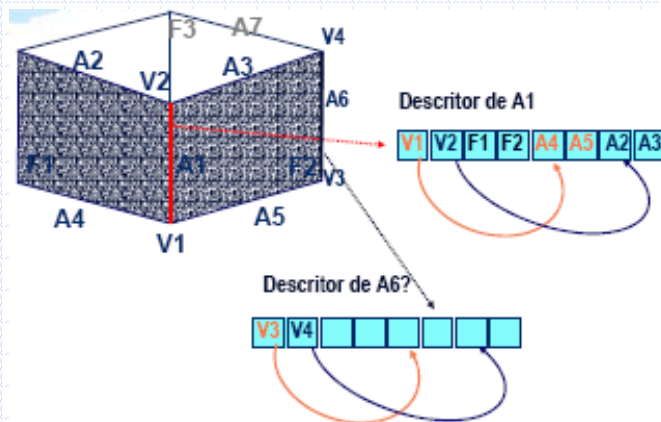
- ◆ A ArestaAlada (Winged-Edge)
- ◆ O Vértice tem um ponteiro para uma de suas ArestasAladas
- ◆ A Face tem um ponteiro para uma de suas ArestasAladas
- ◆ Convenção: indo de N para P, a face P está a direita



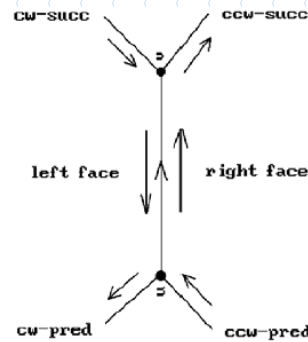
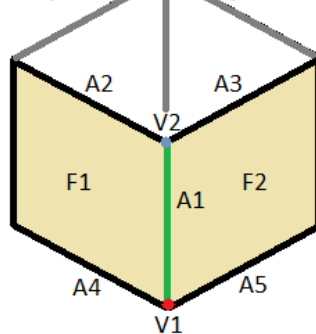
Winged-Edge



Exemplo de Winged-Edge



Winged-Edge



Aresta	Vértice1	Vértice2	Face esq.	Face dir.	CW-Succ	CCW-Succ	CW-Pred	CCW-Pred
A1	V1	V2	F1	F2	A2	A3	A4	A5

Vi

Vf

Fa

Vb

Ponteiros para as outras arestas que saem de Vi e

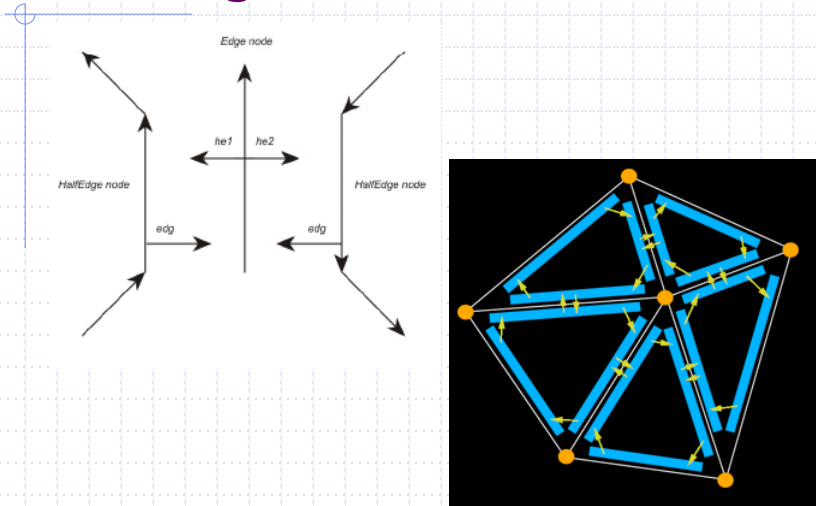


Half-Edge

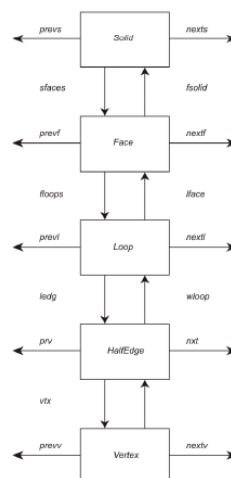
- ◆ Um pouco mais sofisticada que W-E
- ◆ Permite todas as adjacências e outras requisições a serem feitas em tempo constante.
 - Tempo constante para cada pedaço de informação: quando calculando todas as arestas adjacentes a um vértice, a operação cresce linear com o numero de arestas adjacentes ao vértice mas, um tempo constante por aresta.
- ◆ Tamanho finito para as estruturas de faces, arestas e vértices, e compacto
- ◆ Contempla algumas redundâncias



Half-Edge



Half-Edge* Object



H-E versus W-E

- ◆ Half-Edges se restringe a representar superfícies manifold o que a torna inútil para certas aplicações
- ◆ Winged-Edge, com algumas modificações, é capaz de sobrepor algumas das limitações do H-E