

# Processamento de Malhas Poligonais

*Tópicos Avançados em Computação Visual e Interfaces I*

*Prof.: Marcos Lage*

*[www.ic.uff.br/~mlage](http://www.ic.uff.br/~mlage)*

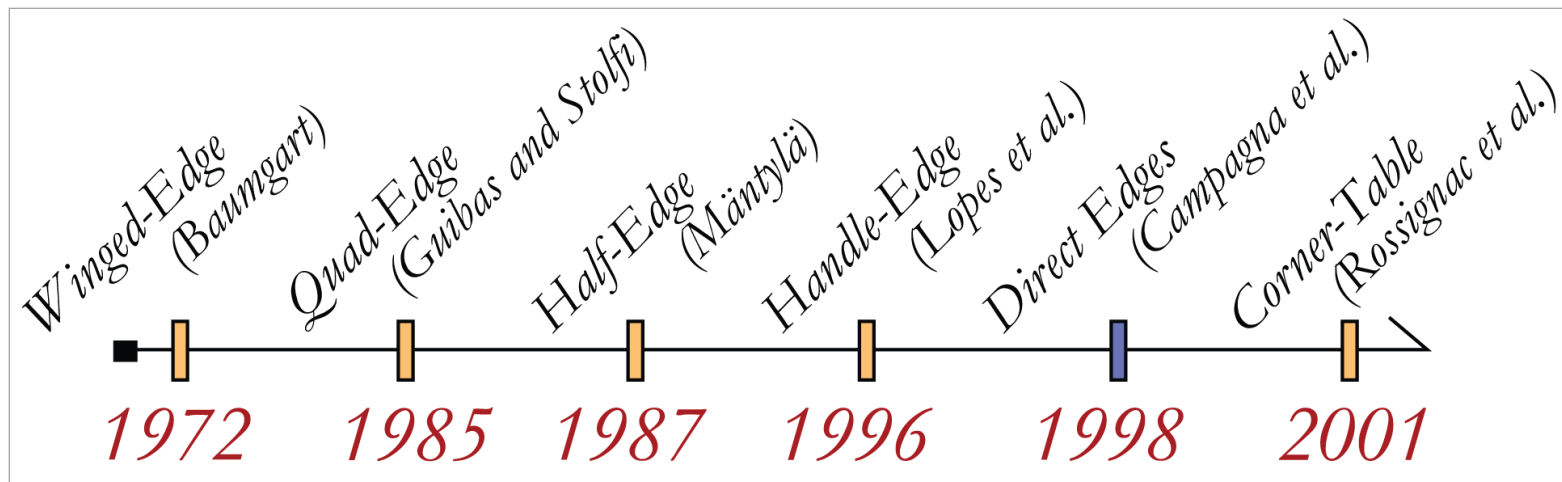
*[mlage@ic.uff.br](mailto:mlage@ic.uff.br)*

*Conteúdo:* Notas de Aula

# Estruturas de Dados Topológicas

## *Introdução*

## Linha do tempo



## Ideias Gerais

De forma geral, podemos representar uma superfície poliédrica através de uma *estrutura de grafos*.

## Ideias Gerais

De forma geral, podemos representar uma superfície poliédrica através de uma *estrutura de grafos*.

Existem *dois grafos* associados à uma superfície poliédrica:

1. *Grafo de vértices*: Induzido pelos vértices e arestas da superfície.

## Ideias Gerais

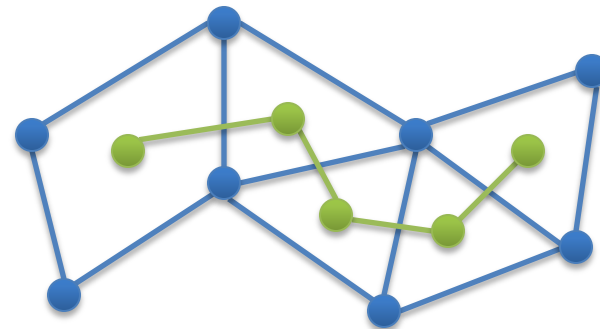
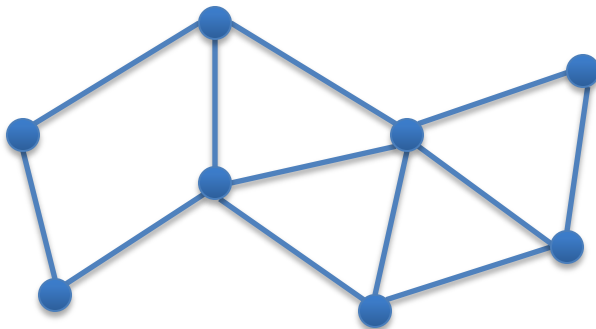
De forma geral, podemos representar uma superfície poliédrica através de uma *estrutura de grafos*.

Existem *dois grafos* associados à uma superfície poliédrica:

1. *Grafo de vértices*: Induzido pelos vértices e arestas da superfície.
2. *Grafo dual*: Grafo cujos vértices representam as faces, e as arestas representam faces adjacentes.

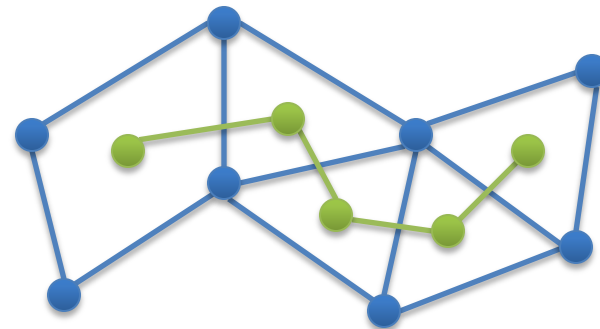
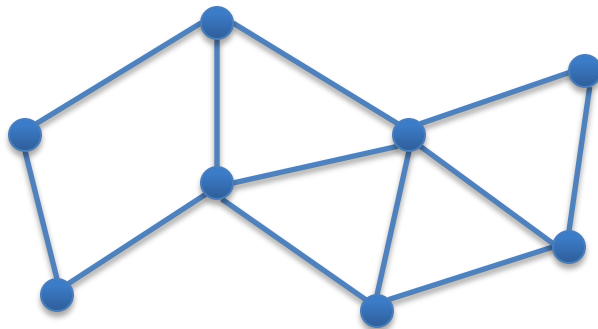
## Ideias Gerais

1. *Grafo de vértices*: Induzido pelos vértices e arestas da superfície.
2. *Grafo dual*: Grafo cujos vértices representam as faces, e as arestas representam faces adjacentes.



## Ideias Gerais

O estudo de Estruturas de Dados Topológicas consiste em desenvolver *formas eficientes de codificar os grafos* associados.





## Ideias Gerais

A representação de uma superfície pode ser vista como um *banco de dados geométrico e topológico*.

## Ideias Gerais

A representação de uma superfície pode ser vista como um *banco de dados geométrico e topológico*.

É comum efetuar certos *tipos de consulta* sobre propriedades geométricas e topológicas da superfície:

## Ideias Gerais

A representação de uma superfície pode ser vista como um *banco de dados geométrico e topológico*.

É comum efetuar certos *tipos de consulta* sobre propriedades geométricas e topológicas da superfície:

1. Achar todas as arestas que incidem em um vértice.
2. Achar todos os polígonos que compartilham uma aresta ou um vértice.
3. Achar as arestas que delimitam um polígono.
4. Visualizar a superfície.
5. ...

## Ideias Gerais

A representação de uma superfície pode ser vista como um banco de dados.

A *escolha da codificação* está intimamente ligada ao conjunto de operações que se deseja realizar.

## Ideias Gerais

A representação de uma superfície pode ser vista como um banco de dados.

A *escolha da codificação* está intimamente ligada ao conjunto de operações que se deseja realizar.

As estruturas de dados para malhas de polígonos mais aceitas pela comunidade científica são as baseadas em semi-arestas (half-edges).

# Estruturas de Dados Topológicas

Definições preliminares

*Half-Edge Clássica*

Half-Edge

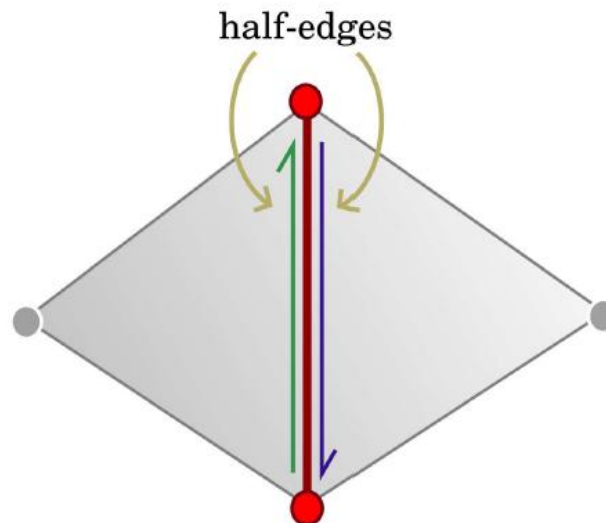
Definição de  
Half-Edge

Uma *half-edge* é uma aresta dotada de uma orientação induzida por um de seus triângulos incidentes.

Half-Edge

Definição de  
Half-Edge

Uma *half-edge* é uma aresta dotada de uma orientação induzida por um de seus triângulos incidentes.





# Estruturas de Dados Topológicas

Half-Edge

Estrutura de  
Dados

A estrutura de dados *half-edge* define três tipos de estruturas/classes:

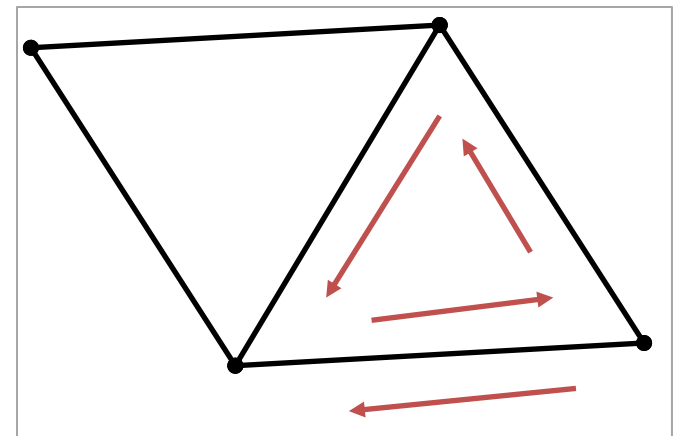
Half-Edge

Estrutura de  
Dados

A estrutura de dados *half-edge* define três tipos de estruturas/classes:

*Vértice*:

Guarda um ponteiro para uma das half-edges com origem no vértice.



Half-Edge

Estrutura de  
Dados

A estrutura de dados *half-edge* define três tipos de estruturas/classes:

**Vértice:**

Guarda um ponteiro para uma das half-edges com origem no vértice.

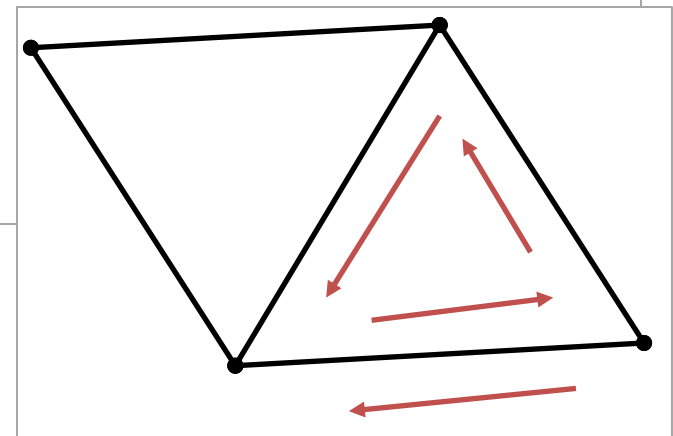
**Half-Edge:**

Guarda um ponteiro para a half-edge seguinte na face.

Guarda um ponteiro para a half-edge oposta.

Guarda um ponteiro para a face.

Guarda um ponteiro para o vértice de origem.



Half-Edge

Estrutura de  
Dados

A estrutura de dados *half-edge* define três tipos de estruturas/classes:

### *Vértice:*

Guarda um ponteiro para uma das half-edges com origem no vértice.

### *Half-Edge:*

Guarda um ponteiro para a half-edge seguinte na face.

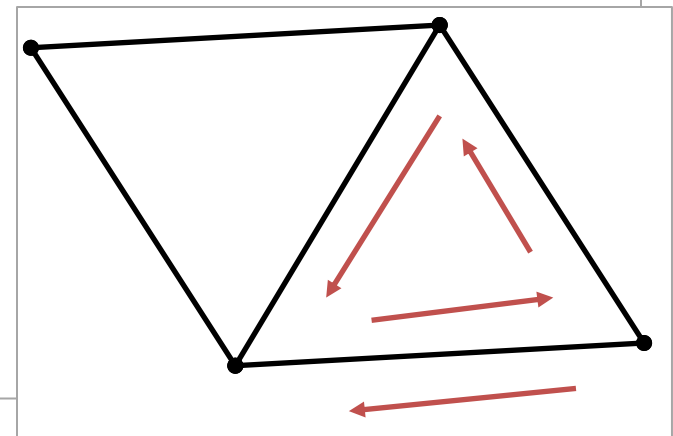
Guarda um ponteiro para a half-edge oposta.

Guarda um ponteiro para a face.

Guarda um ponteiro para o vértice de origem.

### *Face:*

Guarda um ponteiro para uma de suas half-edges.



Half-Edge

Estrutura de  
Dados

A estrutura de dados *half-edge* define três tipos de estruturas/classes:

```
class H_Edge
{
    Vertex vOrig;
    H_Edge eTwin;
    H_Edge eNext;
    Face f;
}
```

```
class Vertex
{
    Point3D p;
    H_Edge hEdge;
}
```

```
class Face
{
    H_Edge hEdge;
}
```

Half-Edge

Estrutura de  
Dados

A estrutura de dados *half-edge* define três tipos de estruturas/classes:

```
class H_Edge
{
    Vertex vOrig;
    H_Edge eTwin;
    H_Edge eNext;
    Face f;
}
```

```
class Vertex
{
    Point3D p;
    H_Edge hEdge;
}
```

```
class Face
{
    H_Edge hEdge;
}
```

## Observações:

Para um mesmo Vertex podem existir diversos H\_Edge, porém, basta guardar um deles com origem no Vertex.

Half-Edge

Estrutura de  
Dados

A estrutura de dados *half-edge* define três tipos de estruturas/classes:

```
class H_Edge
{
    Vertex vOrig;
    H_Edge eTwin;
    H_Edge eNext;
    Face f;
}
```

```
class Vertex
{
    Point3D p;
    H_Edge hEdge;
}
```

```
class Face
{
    H_Edge hEdge;
}
```

## Observações:

Na face, basta guardar uma de suas H\_Edge

# Estruturas de Dados Topológicas

Definições preliminares

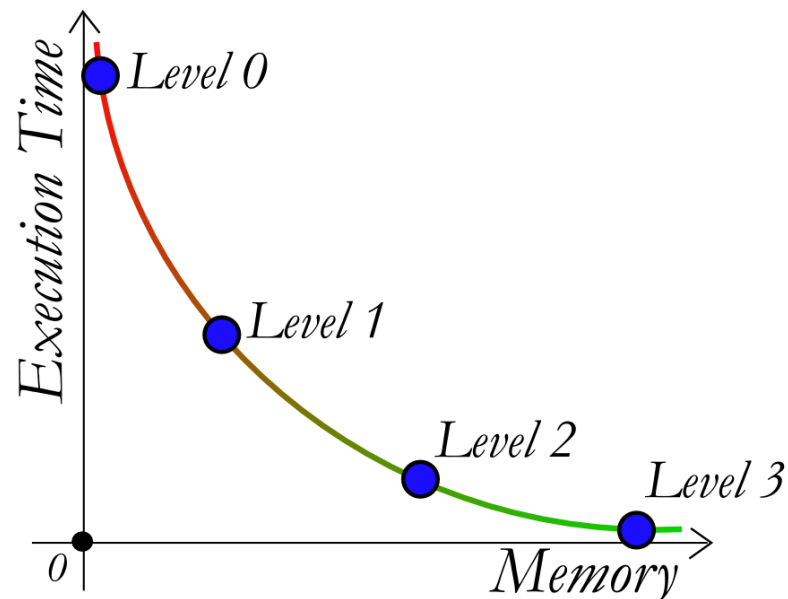
Half-Edge Clássica

*Half-Edge Compacta*



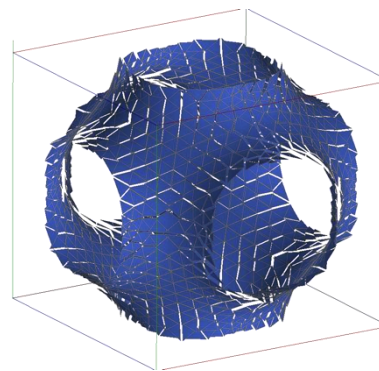
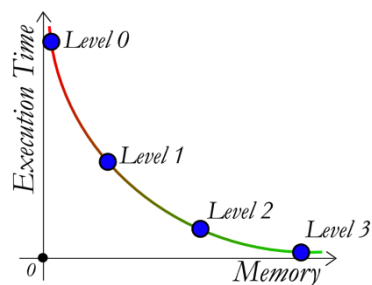
Half-Edge Compacta

Processamento  
Vs Memória



## Half-Edge Compacta

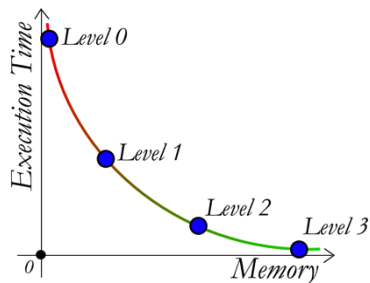
Processamento  
Vs Memória



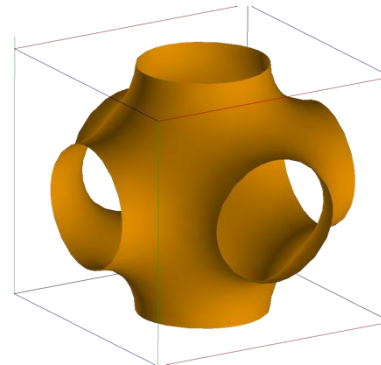
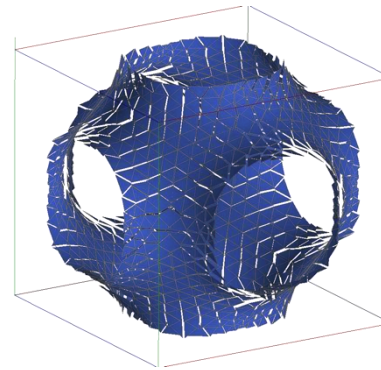
**Nível 0:** Rendering.

## Half-Edge Compacta

Processamento  
Vs Memória

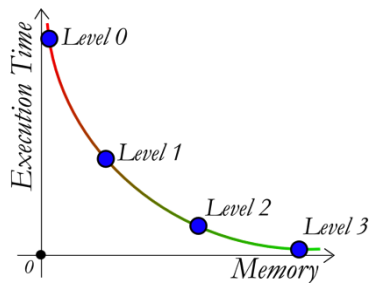


**Nível 0:** Rendering.  
**Nível 1:** Vizinhança.

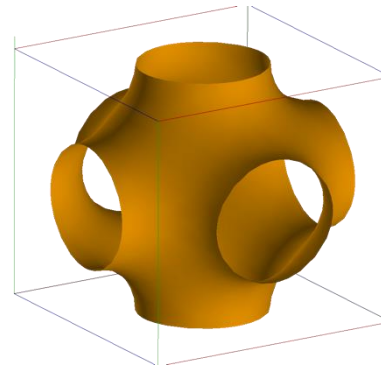
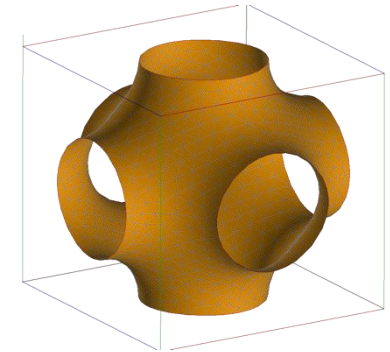
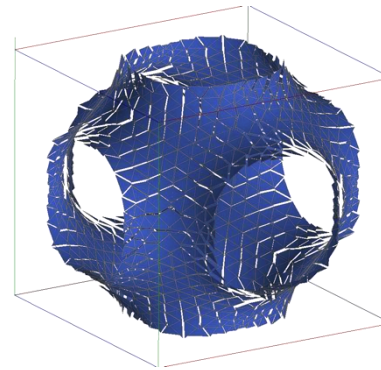


## Half-Edge Compacta

Processamento  
Vs Memória

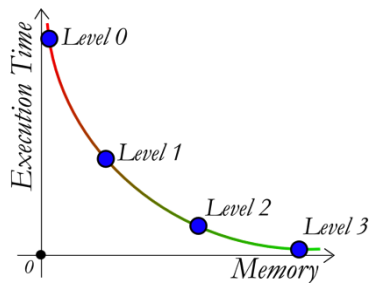


**Nível 0:** Rendering.  
**Nível 1:** Vizinhança.  
**Nível 2:** Simplexos.

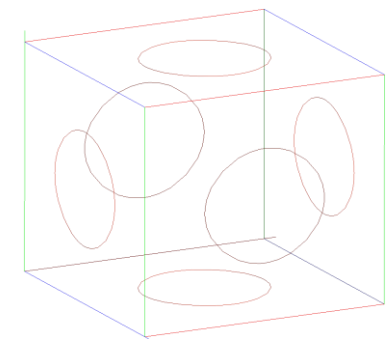
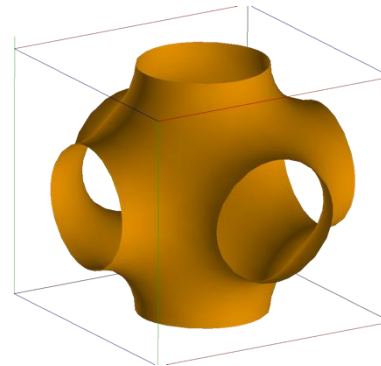
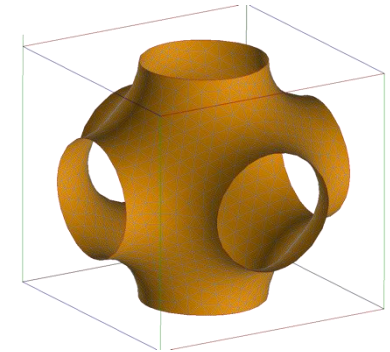
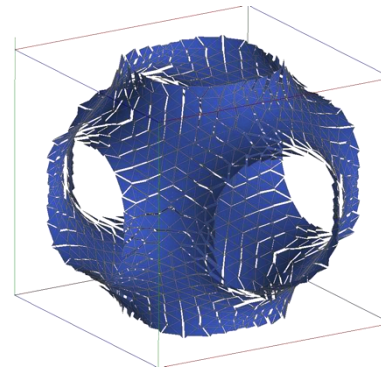


## Half-Edge Compacta

Processamento  
Vs Memória

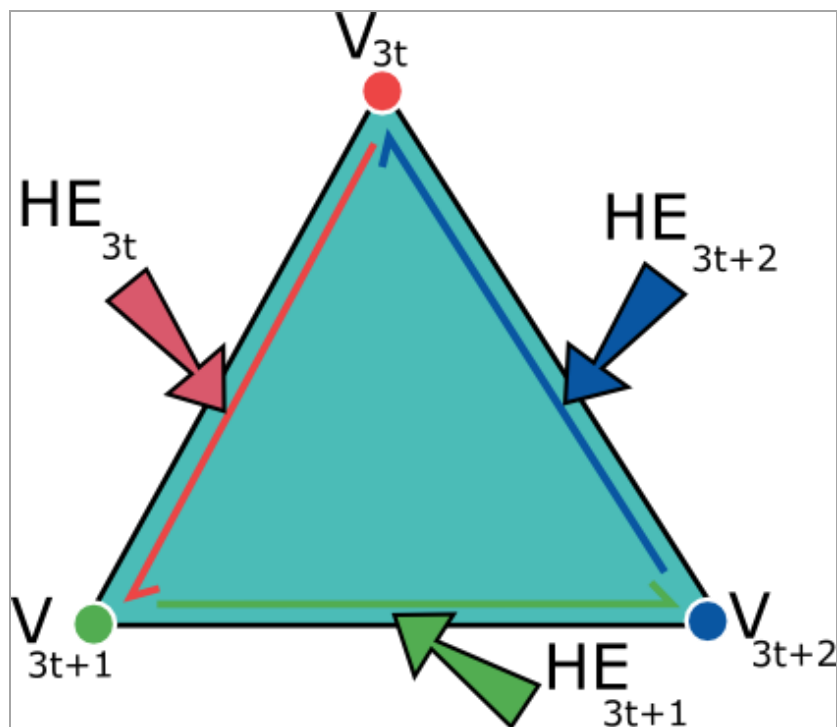


**Nível 0:** Rendering.  
**Nível 1:** Vizinhança.  
**Nível 2:** Simplexos.  
**Nível 3:** Bordo.



Half-Edge Compacta

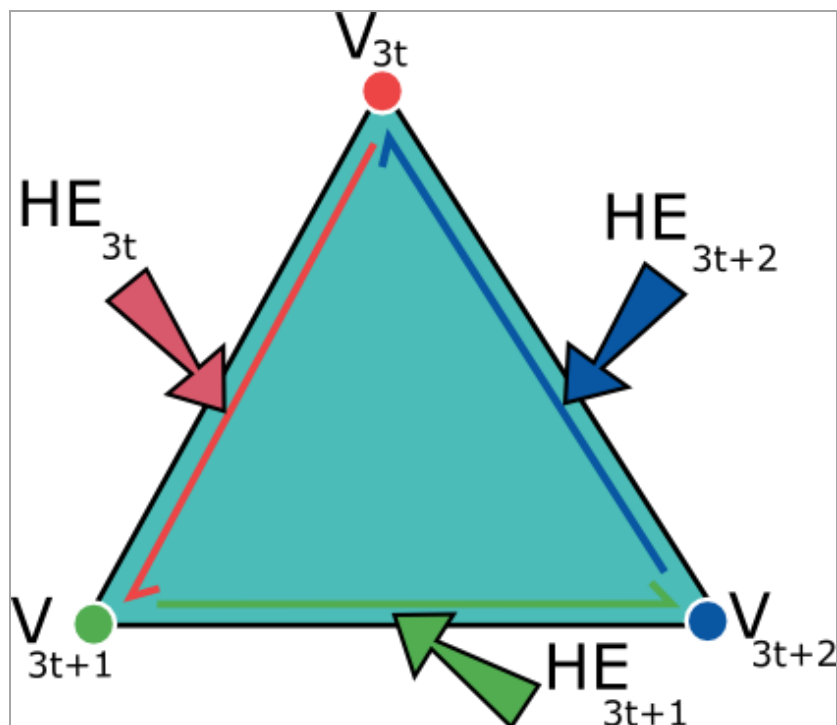
Indexação  
das HEs



Sai ponteiro, entra *aritmética inteira*.

## Half-Edge Compacta

Indexação  
das HEs

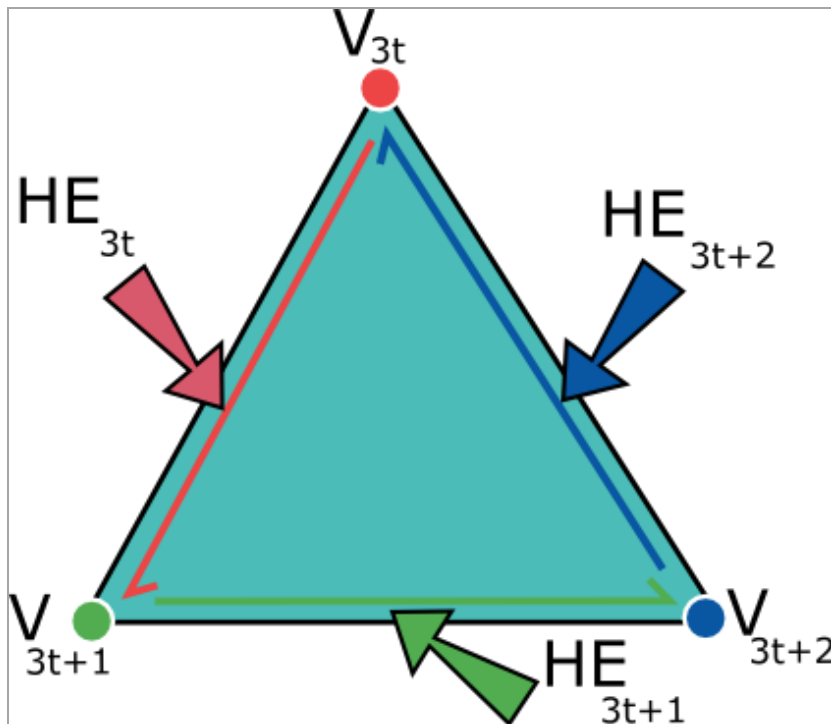


Sai ponteiro, entra *aritmética inteira*.

As half-edges e os polígonos são *indexados* por números inteiros não-negativos.

## Half-Edge Compacta

Indexação  
das HEs



Sai ponteiro, entra *aritmética inteira*.

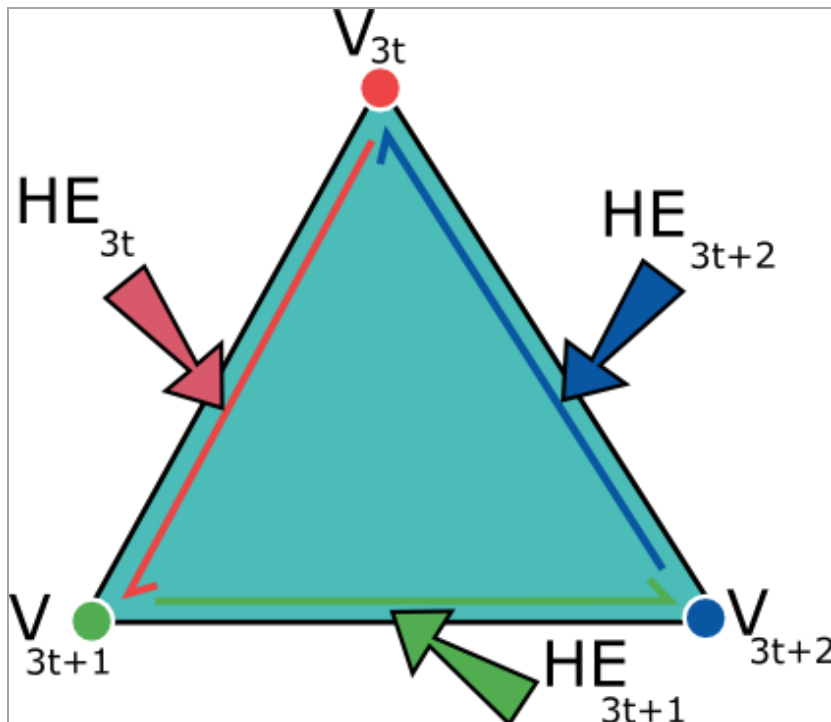
As half-edges e os polígonos são *indexados* por números inteiros não-negativos.

Cada triângulo é representado por 3 half-edges de índices consecutivos, que formam seu bordo orientado.



## Half-Edge Compacta

Indexação  
das HEs



Sai ponteiro, entra *aritmética inteira*.

As half-edges e os polígonos são *indexados* por números inteiros não-negativos.

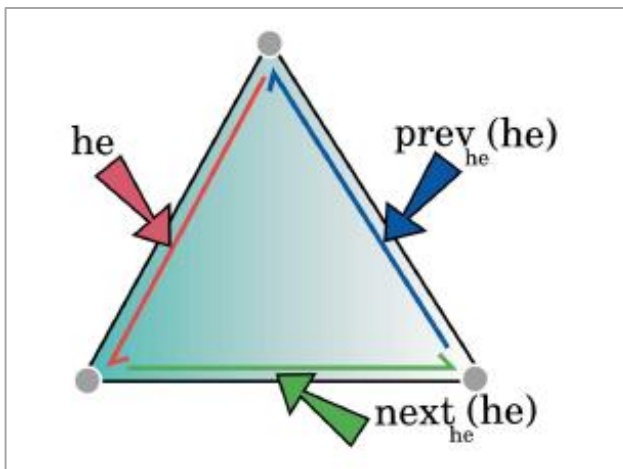
Cada triângulo é representado por 3 half-edges de índices consecutivos, que formam seu bordo orientado.

Ex:

$T_0 \rightarrow he_0 he_1 he_2$      $T_1 \rightarrow he_3 he_4 he_5$

Half-Edge Compacta

Aritmética  
Inteira



$$\text{Trig}(\text{he}) := \lfloor \text{he}/3 \rfloor$$

$$\text{Next}_{\text{he}}(\text{he}) := 3 * \text{Trig}(\text{he}) + (\text{he} + 1) \% 3$$

$$\text{Prev}_{\text{he}}(\text{he}) := 3 * \text{Trig}(\text{he}) + (\text{he} + 2) \% 3$$

Half-Edge Compacta

Tabelas  
G & V

Na **CHE** as informações geométricas são armazenadas em uma tabela, denotada por **G**, cujos índices variam de 0 ao número de vértices da malha.

Half-Edge Compacta

Tabelas  
G & V

Na **CHE** as informações geométricas são armazenadas em uma tabela, denotada por **G**, cujos índices variam de 0 ao número de vértices da malha.

Na **CHE** as half-edges são armazenadas em uma tabela, denotada por **V**, cujos índices variam de 0 à três vezes o número de triângulos da malha.

Half-Edge Compacta

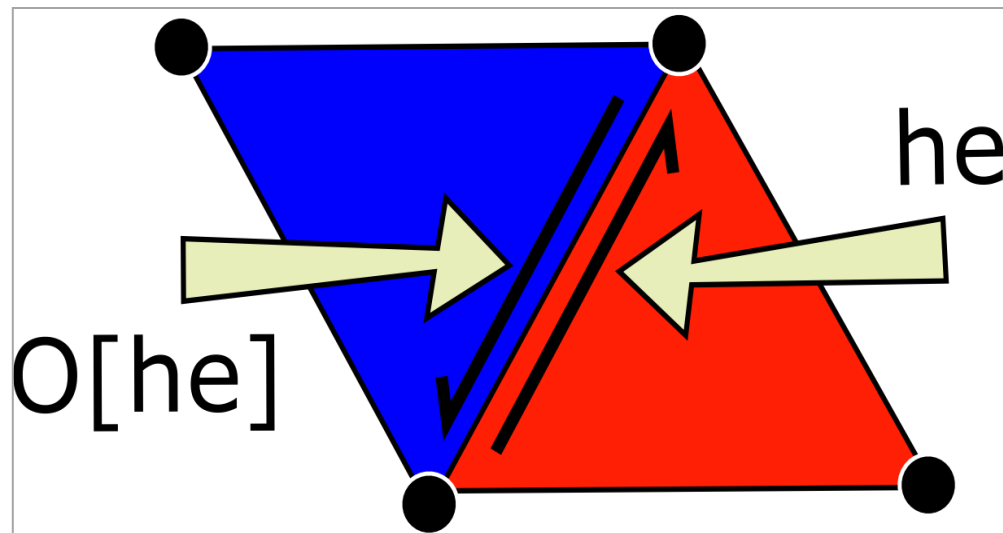
Half-Edges  
Opostas

Duas *half-edges* são *opostas* quando compartilham um mesmo vértice mas possuem orientação oposta.

Half-Edge Compacta

Half-Edges  
Opostas

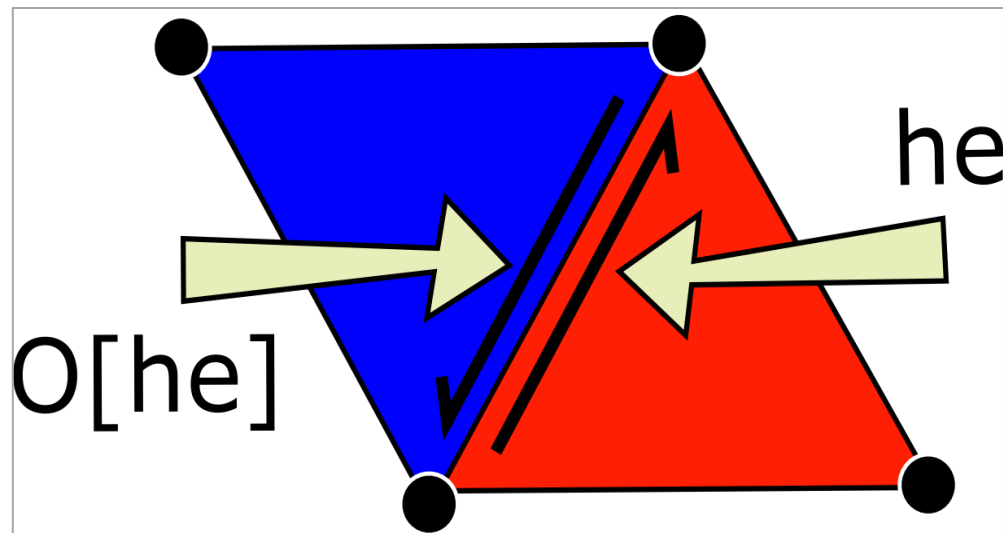
Duas *half-edges* são *opostas* quando compartilham um mesmo vértice mas possuem orientação oposta.



Half-Edge Compacta

Tabela  
O

Duas *half-edges* são *opostas* quando compartilham um mesmo vértice mas possuem orientação oposta.



Na *CHE* as half-edges opostas são armazenadas em uma tabela, denotada por *O*, cujos índices variam de 0 à três vezes o número de triângulos da malha.

Half-Edge Compacta

Tabela  
Auxiliares

Na **CHE** outras tabelas podem ser criadas para que haja representação explícita de simplexos.



Half-Edge Compacta

Tabela  
Auxiliares

Na **CHE** outras tabelas podem ser criadas para que haja representação explícita de simplexos.

**Ex1:** Tabela **VH** que armazena uma das arestas incidentes a um vértice, cujos índices vão de 0 ao número de vértices da malha.

Half-Edge Compacta

Tabela  
Auxiliares

Na **CHE** outras tabelas podem ser criadas para que haja representação explícita de simplexos.

**Ex1:** Tabela **VH** que armazena uma das arestas incidentes a um vértice, cujos índices vão de 0 ao número de vértices da malha.

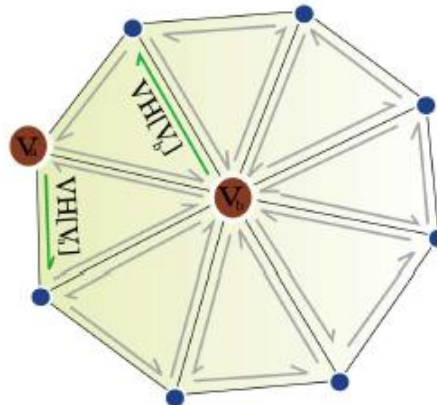
**Ex2:** Tabela **EH** que armazena uma das half-edges que incidem sobre as arestas, cujos índices vão de 0 ao número de arestas da malha.

Half-Edge Compacta

Tabela  
Auxiliares

Na **CHE** outras tabelas podem ser criadas para que haja representação explícita de simplexos.

**OBS: Quando o simplexo representado for de bordo, convém armazenar uma half-edge de bordo nas tabelas auxiliares...**



# Estruturas de Dados Topológicas

Definições preliminares

Half-Edge Clássica

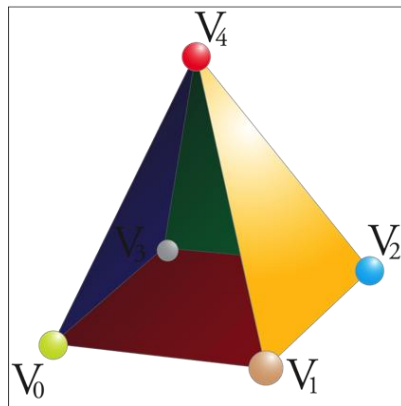
Half-Edge Compacta

*Exemplo concreto*

Half-Edge Compacta

Exemplo  
Concreto

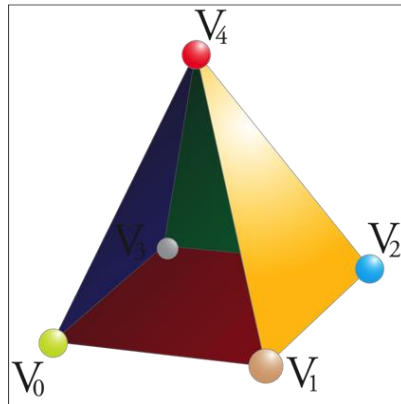
complexo  
dimensão 2



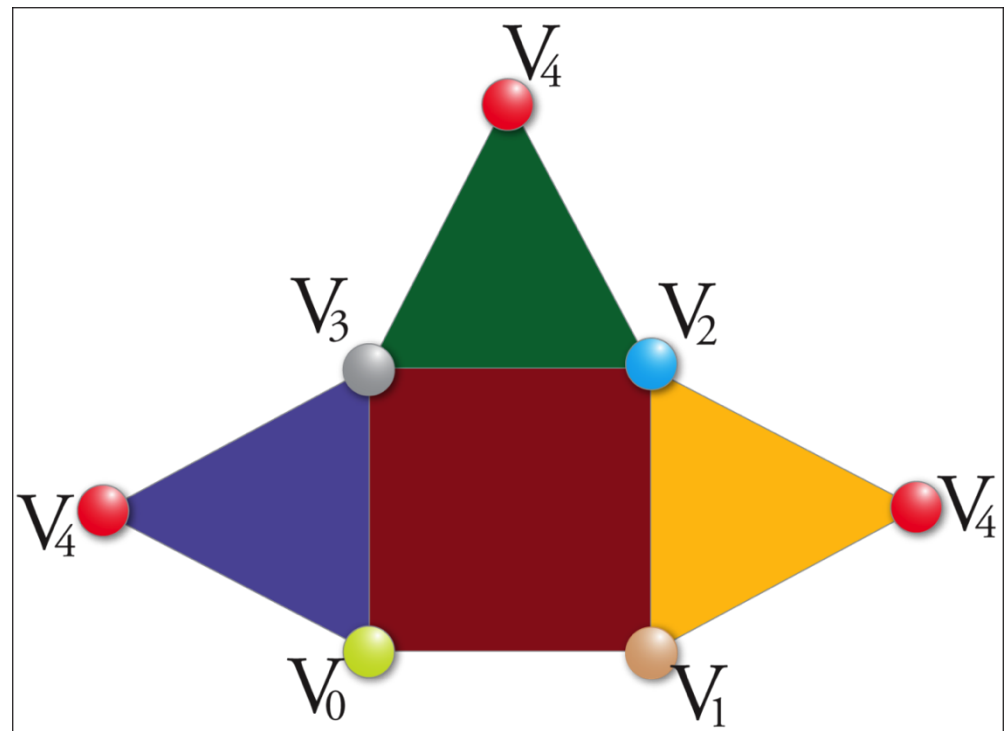
Half-Edge Compacta

Exemplo  
Concreto

complexo  
dimensão 2



planificação

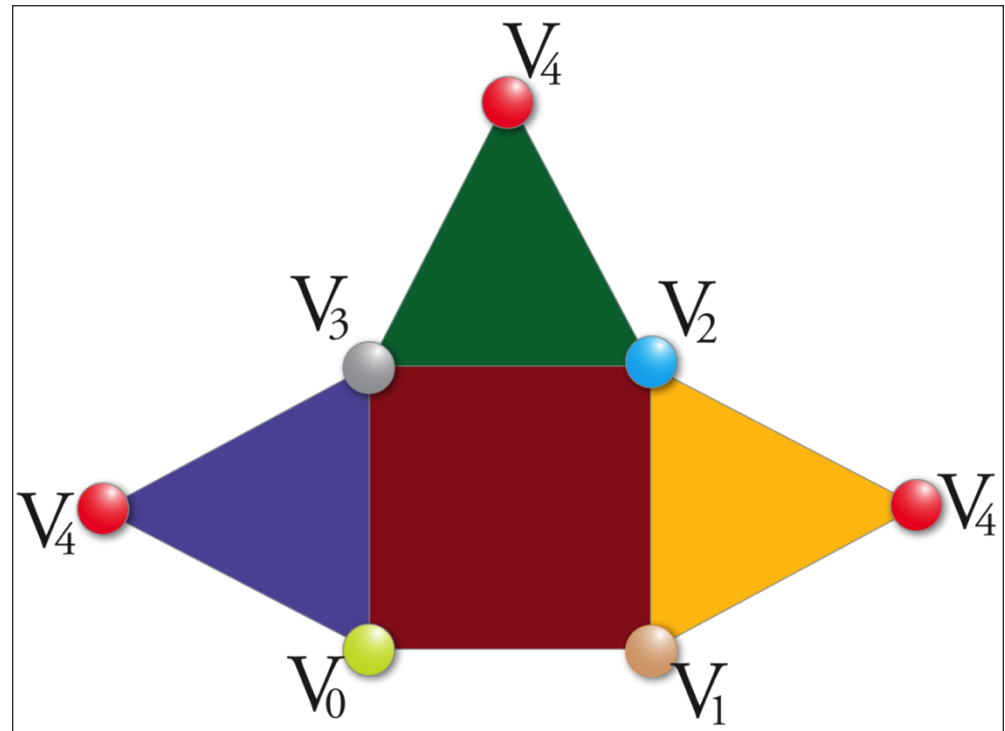


Half-Edge Compacta

Exemplo  
Concreto

Geometry Table

$V_0$	$P_0$	$\vec{N}_0$ (...)
$V_1$	$P_1$	$\vec{N}_1$ (...)
$V_2$	$P_2$	$\vec{N}_2$ (...)
$V_3$	$P_3$	$\vec{N}_3$ (...)
$V_4$	$P_4$	$\vec{N}_4$ (...)



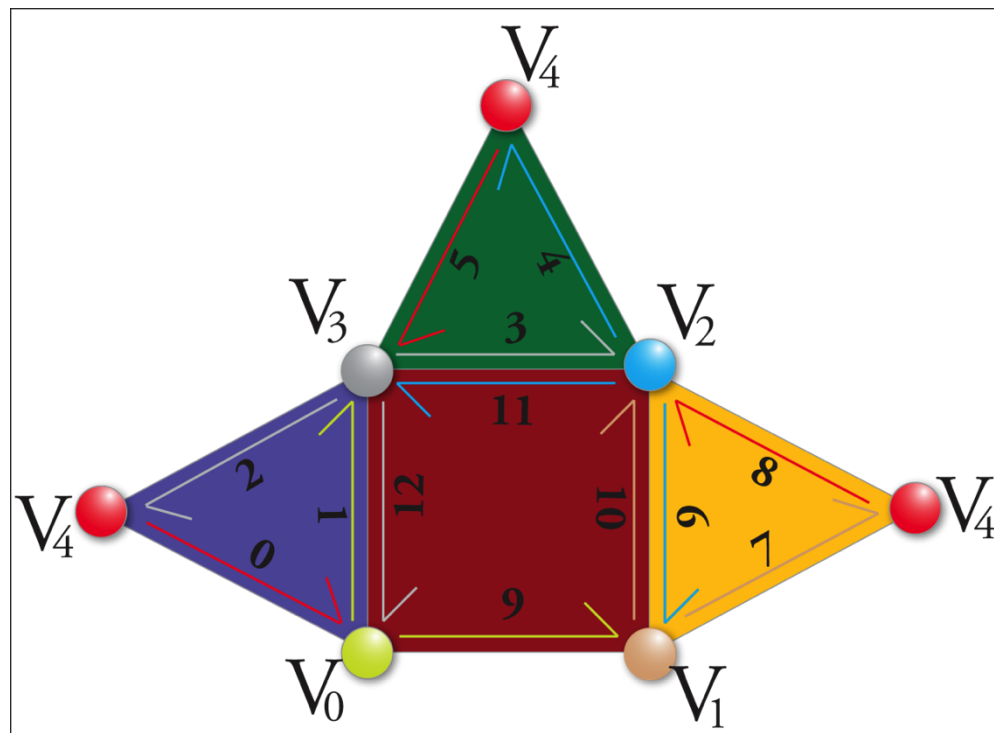
Half-Edge Compacta

Exemplo  
Concreto

Vertex Table

$h_0$	$h_1$	$h_2$	$h_3$	$h_4$	$h_5$	$h_6$	$h_7$	$h_8$
4	0	3	3	2	4	2	1	4

$h_9$	$h_{10}$	$h_{11}$	$h_{12}$
0	1	2	3





Half-Edge Compacta

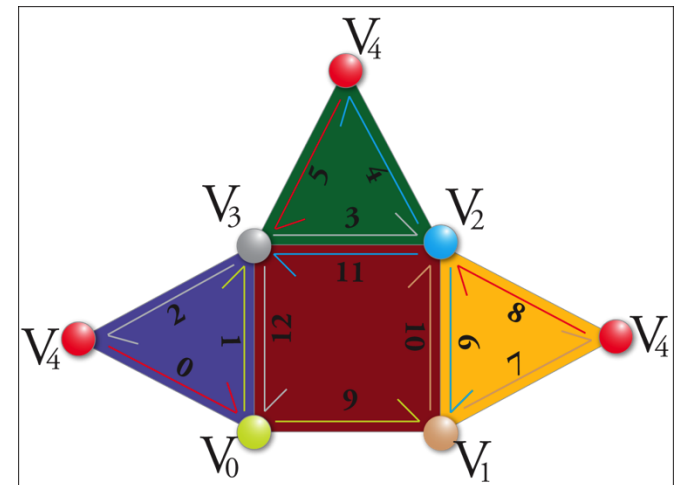
Exemplo  
Concreto

Vertex Table

$h_0$	$h_1$	$h_2$	$h_3$	$h_4$	$h_5$	$h_6$	$h_7$	$h_8$
4	0	3	3	2	4	2	1	4

$h_9$	$h_{10}$	$h_{11}$	$h_{12}$
0	1	2	3

*Estrela de  $V_3$*



Half-Edge Compacta

Exemplo  
Concreto

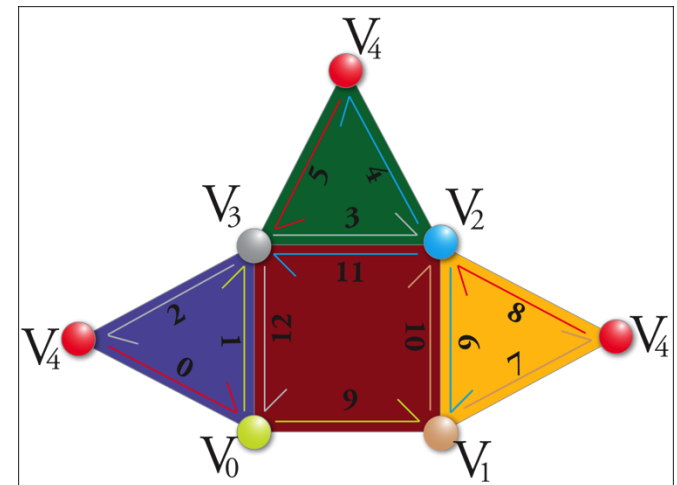
Vertex Table

$h_0$	$h_1$	$h_2$	$h_3$	$h_4$	$h_5$	$h_6$	$h_7$	$h_8$
4	0	3	3	2	4	2	1	4

$h_9$	$h_{10}$	$h_{11}$	$h_{12}$
0	1	2	3

*Estrela de  $V_3$*

2



Half-Edge Compacta

Exemplo  
Concreto

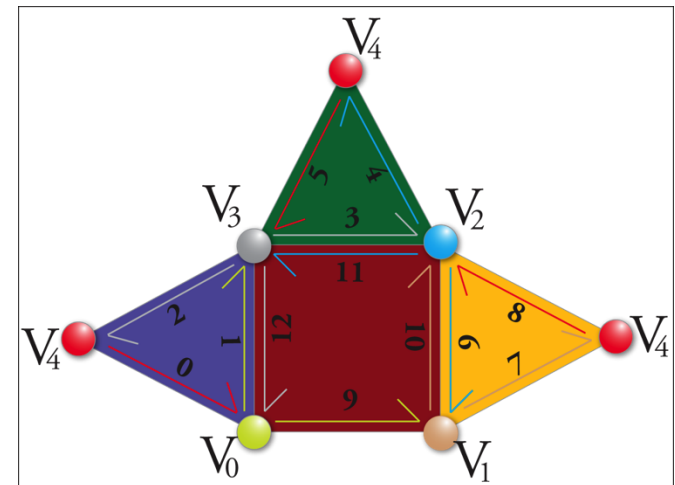
Vertex Table

$h_0$	$h_1$	$h_2$	$h_3$	$h_4$	$h_5$	$h_6$	$h_7$	$h_8$
4	0	3	3	2	4	2	1	4

$h_9$	$h_{10}$	$h_{11}$	$h_{12}$
0	1	2	3

*Estrela de  $V_3$*

2 3



Half-Edge Compacta

Exemplo  
Concreto

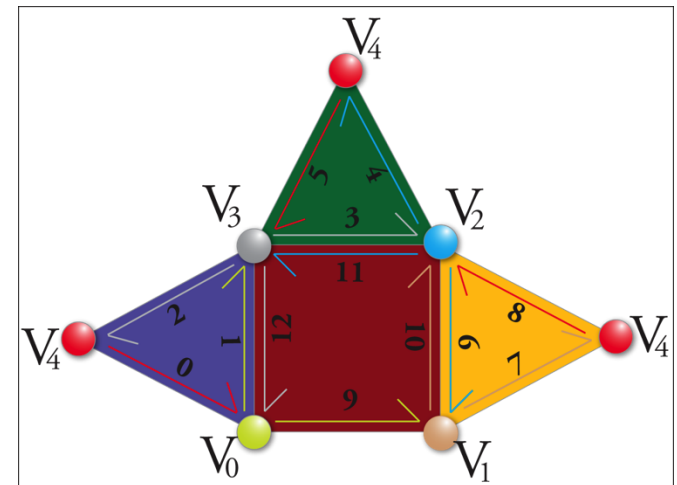
Vertex Table

$h_0$	$h_1$	$h_2$	$h_3$	$h_4$	$h_5$	$h_6$	$h_7$	$h_8$
4	0	3	3	2	4	2	1	4

$h_9$	$h_{10}$	$h_{11}$	$h_{12}$
0	1	2	3

*Estrela de  $V_3$*

2 3 12



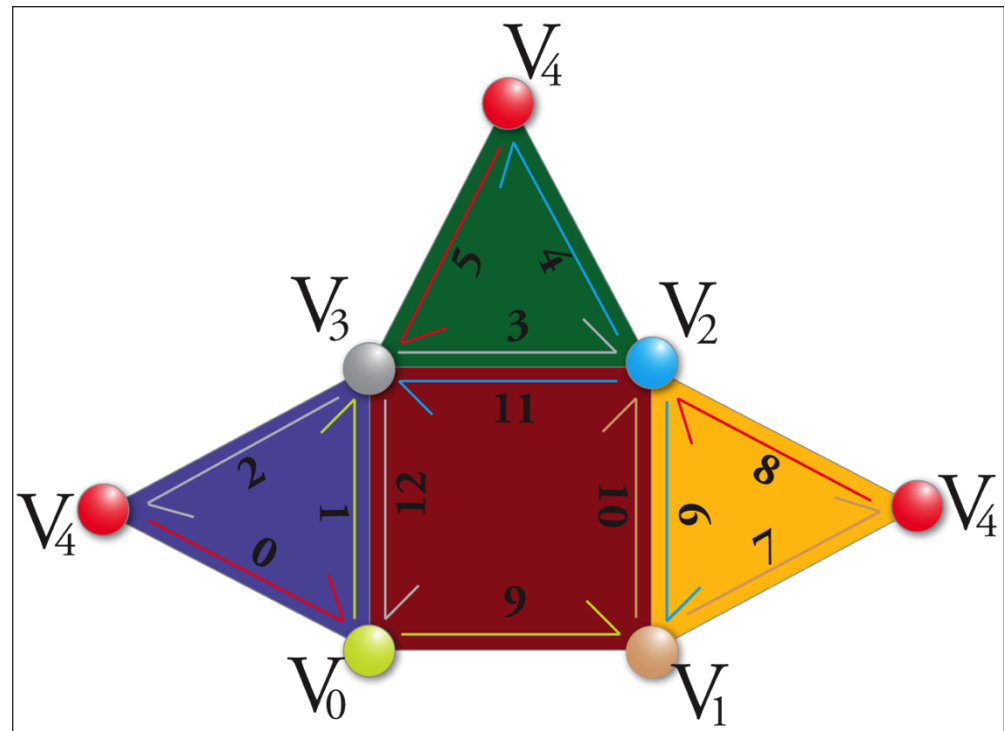
Half-Edge Compacta

Exemplo  
Concreto

Opposite Table

$h_0$	$h_1$	$h_2$	$h_3$	$h_4$	$h_5$	$h_6$	$h_7$	$h_8$
-1	12	5	11	8	2	10	-1	4

$h_9$	$h_{10}$	$h_{11}$	$h_{12}$
-1	6	3	1



## Half-Edge Compacta

Exemplo  
Concreto

### Vertex Table

$h_0$	$h_1$	$h_2$	$h_3$	$h_4$	$h_5$	$h_6$	$h_7$	$h_8$
4	0	3	3	2	4	2	1	4

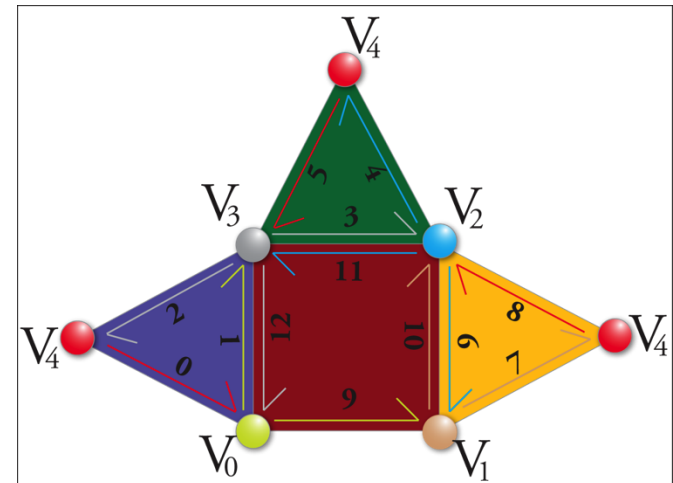
$h_9$	$h_{10}$	$h_{11}$	$h_{12}$
0	1	2	3

### Opposite Table

$h_0$	$h_1$	$h_2$	$h_3$	$h_4$	$h_5$	$h_6$	$h_7$	$h_8$
-1	12	5	11	8	2	10	-1	4

$h_9$	$h_{10}$	$h_{11}$	$h_{12}$
-1	6	3	1

*Estrela de  $V_3$*



## Half-Edge Compacta

Exemplo  
Concreto

### Vertex Table

$h_0$	$h_1$	$h_2$	$h_3$	$h_4$	$h_5$	$h_6$	$h_7$	$h_8$
4	0	3	3	2	4	2	1	4

$h_9$	$h_{10}$	$h_{11}$	$h_{12}$
0	1	2	3

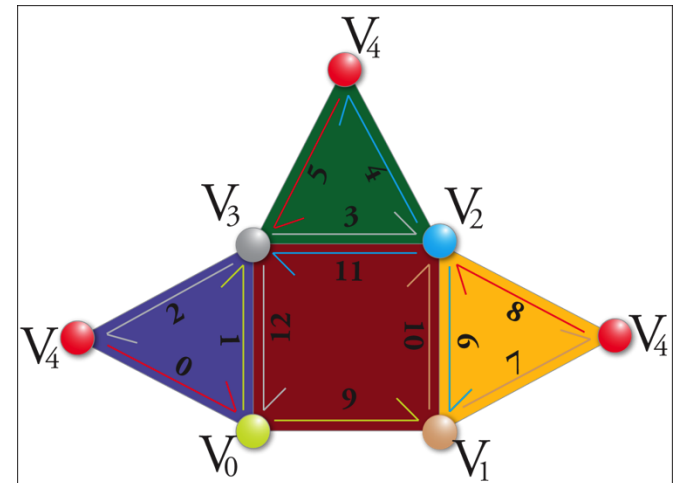
### Opposite Table

$h_0$	$h_1$	$h_2$	$h_3$	$h_4$	$h_5$	$h_6$	$h_7$	$h_8$
-1	12	5	11	8	2	10	-1	4

$h_9$	$h_{10}$	$h_{11}$	$h_{12}$
-1	6	3	1

*Estrela de  $V_3$*

2



## Half-Edge Compacta

Exemplo  
Concreto

### Vertex Table

$h_0$	$h_1$	$h_2$	$h_3$	$h_4$	$h_5$	$h_6$	$h_7$	$h_8$
4	0	3	3	2	4	2	1	4

$h_9$	$h_{10}$	$h_{11}$	$h_{12}$
0	1	2	3

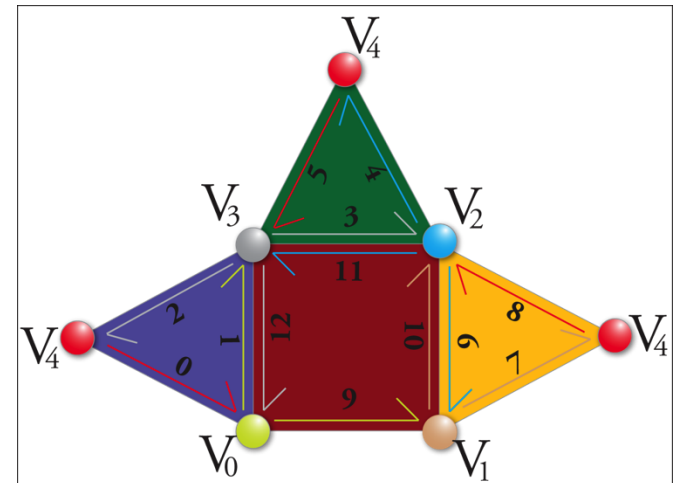
### Opposite Table

$h_0$	$h_1$	$h_2$	$h_3$	$h_4$	$h_5$	$h_6$	$h_7$	$h_8$
-1	12	5	11	8	2	10	-1	4

$h_9$	$h_{10}$	$h_{11}$	$h_{12}$
-1	6	3	1

*Estrela de  $V_3$*

2 3





## Half-Edge Compacta

Exemplo  
Concreto

### Vertex Table

$h_0$	$h_1$	$h_2$	$h_3$	$h_4$	$h_5$	$h_6$	$h_7$	$h_8$
4	0	3	3	2	4	2	1	4

$h_9$	$h_{10}$	$h_{11}$	$h_{12}$
0	1	2	3

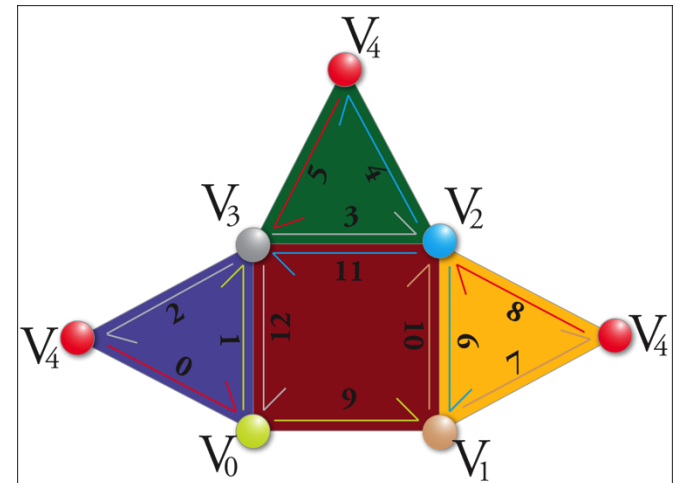
### Opposite Table

$h_0$	$h_1$	$h_2$	$h_3$	$h_4$	$h_5$	$h_6$	$h_7$	$h_8$
-1	12	5	11	8	2	10	-1	4

$h_9$	$h_{10}$	$h_{11}$	$h_{12}$
-1	6	3	1

*Estrela de  $V_3$*

2 3 12



Half-Edge Compacta

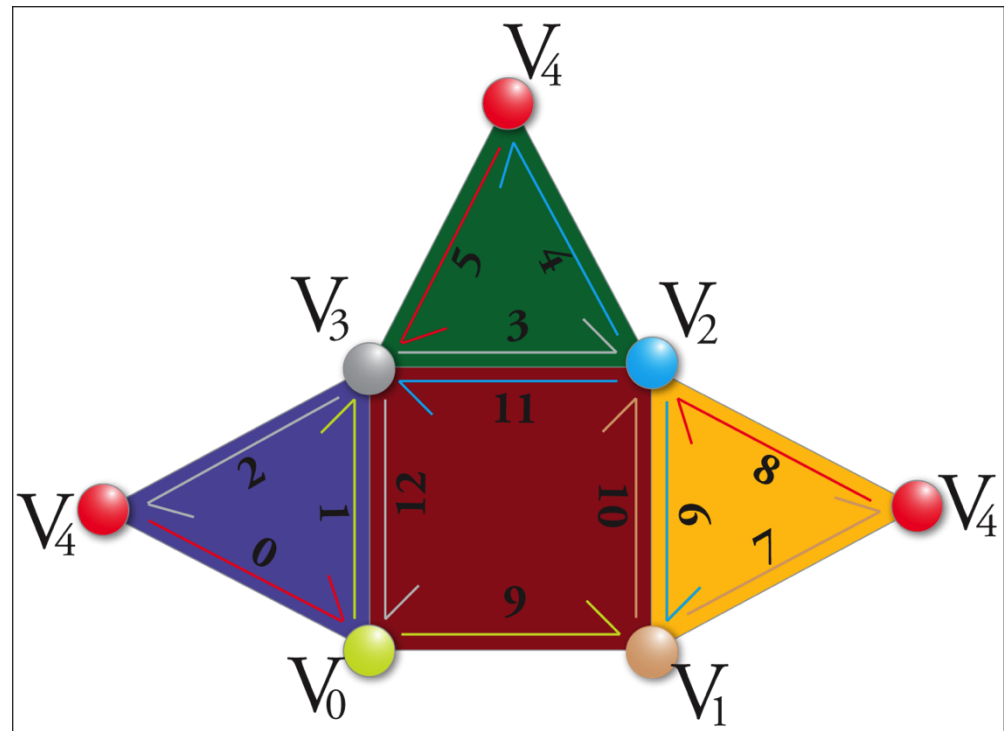
Exemplo  
Concreto

Extra Vertex Table

$V_0$	$V_1$	$V_2$	$V_3$	$V_4$
9	7	4	2	0

Edge Table

$E_0$	$E_1$	$E_2$	$E_3$	$E_4$	$E_5$	$E_6$	$E_7$
0	1	2	3	4	6	7	9

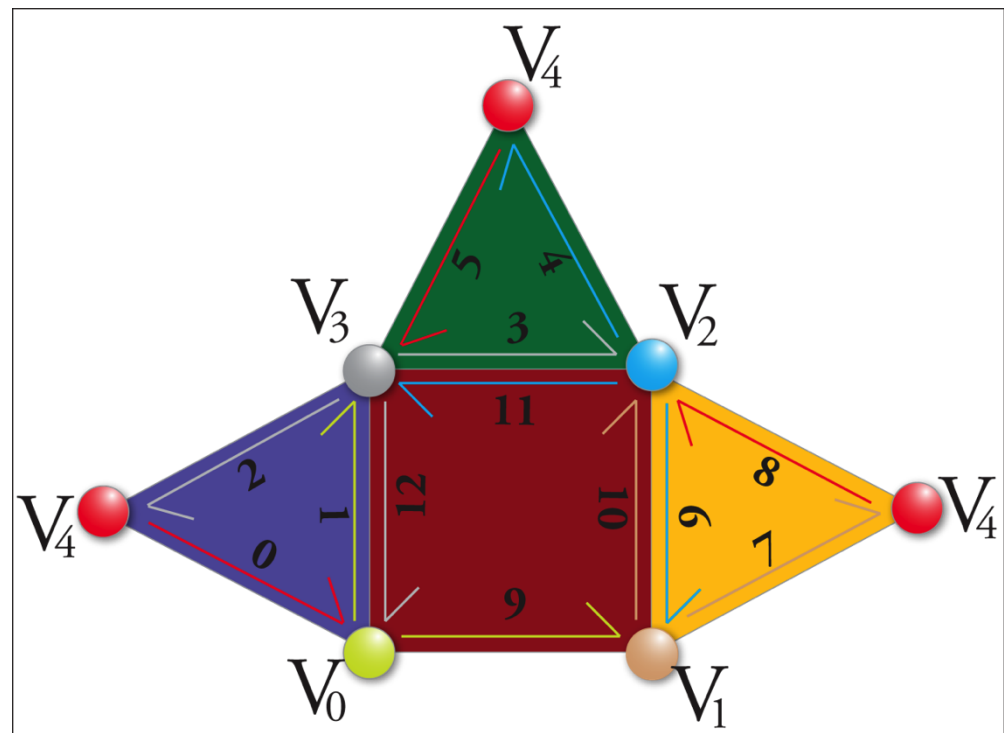


Half-Edge Compacta

Exemplo  
Concreto

Boundary Table

$B_0$	0
-------	---



## Half-Edge Compacta

Código  
Fonte

```
/**
 * Processamento de malhas poligonais.
 * 3ª Gradação IC-UFF 2012.1
 * Classe Che
 * @author Prof. Marcos Lage <mlage@ic.uff.br>
 */

package ic.mlibs.structures;

//file i-o imports
import ic.mlibs.util.Helper;
import java.io.BufferedReader;
import java.io.File;
import java.io.DataOutputStream;
import java.io.FileOutputStream;

//file i-o exceptions imports
import java.io.FileReader;
import java.io.IOException;

//Data structure imports
import java.util.Stack;
import java.util.HashMap;
import java.util.ArrayList;
import java.util.AbstractMap.SimpleEntry;
import java.util.Collections;
import java.util.Scanner;

class Che {

    protected int nvert; //> Número de vértices da malha.
    protected int ntris; //> Número de triângulos da malha.
    protected int nquad; //> Número de quadrângulos da malha.
    protected int ncomp; //> Número de componentes conexas da malha.
    protected int ncurv; //> Número de curvas de bordo da malha.
    protected ArrayList<Point3D> G; //> Coordenada dos vértices da malha.
    protected ArrayList<Integer> VT; //> Índice dos vértices dos trigs da malha.
    protected ArrayList<Integer> VQ; //> Índice dos vértices dos quads da malha.
    protected ArrayList<Integer> OT; //> Container de half-edges vizinhas de trigs.
    protected ArrayList<Integer> OQ; //> Container de half-edges vizinhas de quads.
    protected HashMap
        <Integer, Object> EH; //> Representação de arestas.
    protected ArrayList<Integer> VH; //> Primeiro triângulo da estrela de um vértice.
    protected ArrayList<Integer> C; //> Componente conexa de cada vértice.
    protected ArrayList<Integer> B; //> Half-edge representante de cada curva de bordo.

    //-----
    public Che() {
    }

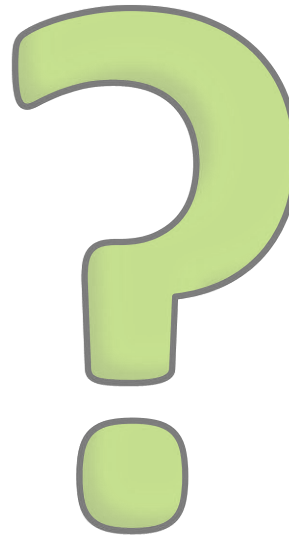
    public Che(String file) {
        readChe(file);
    }

    //-----
    /**
     * Acesso ao número de vértices da malha.
     */
    public int getNvert() {
        return nvert;
    }
}
```

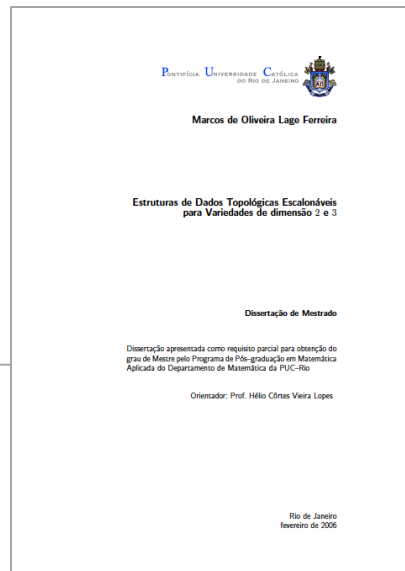


Half-Edge Compacta

Malhas  
Híbridas



## Estruturas de dados compactas



**Material complementar:**  
*Estruturas de dados topológicas escalonáveis para variedades de dimensão 2 e 3*

M. Lage  
M.Sc. Thesis  
2006