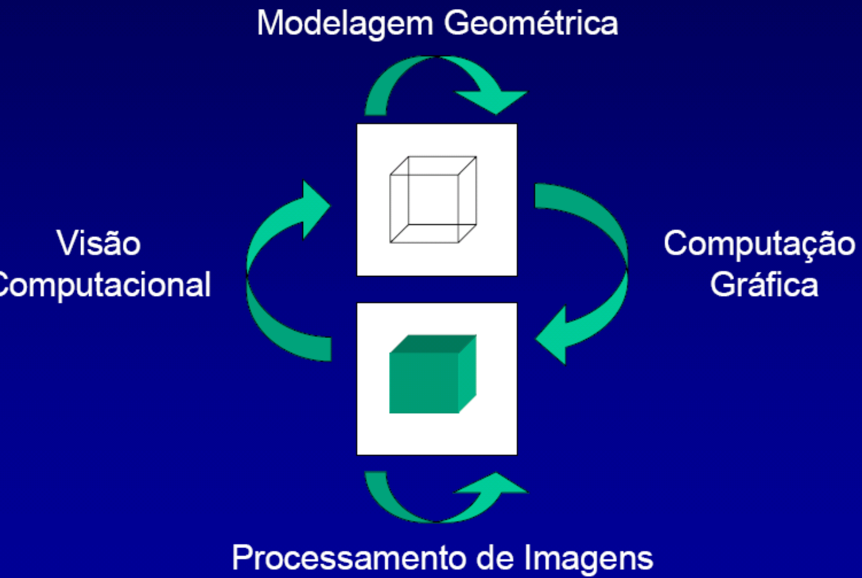
Fundamentos de Processamento Gráfico

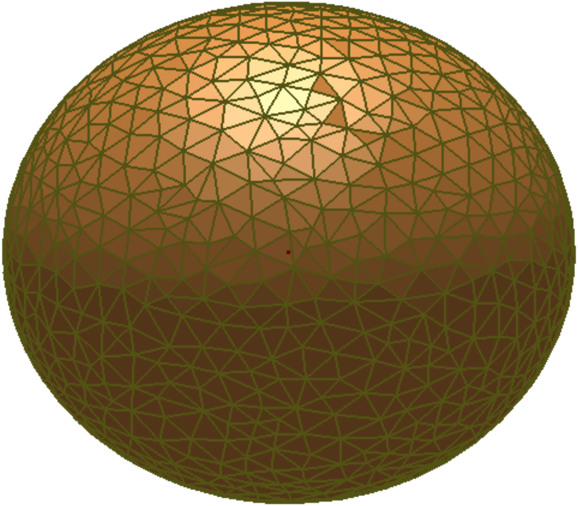
Helton H. Bíscaro ; Fátima Nunes

31 de outubro de 2019

Áreas Correlatas



Estrutruras de Dados



Células e Simplexos

Célula (definição:)

Dado um conjunto de pontos $p_0, p_1, ..., p_k \in \mathbb{R}^n$, a célula gerada por este conjunto é a combinação convexa

$$[p_0, p_1, ..., p_n] = \left\{ \sum_{i=0}^{k} \lambda_i p_i ; \lambda_i \geq 0 ; \sum_{i=0}^{k} \lambda_i = 1 \right\}$$

Células e Simplexos

Célula (definição:)

Dado um conjunto de pontos $\{p_0, p_1, ..., p_k\} \in \mathbb{R}^n$, a célula gerada por este conjunto é a combinação convexa

$$[p_0, p_1, ..., p_n] = \left\{ \sum_{i=0}^{k} \lambda_i p_i; \lambda_i \geq 0; \sum_{i=0}^{k} \lambda_i = 1 \right\}$$

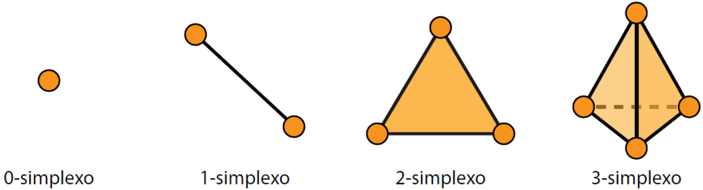
Exemplo

A célula gerada pelos pontos $[p_0, p_1, p_2]$ pode ser um ponto, um segmento de reta, ou um triângulo, de acordo com a relação de dependência linear de $p_1 - p_0, p_2 - p_0$.

Células e Simplexos

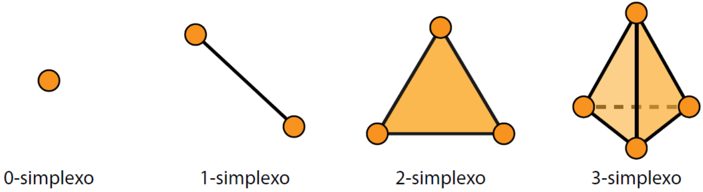
Simplexo (definição:)

Quando um conjunto de pontos $\{p_0, p_1, ..., p_k\} \in \mathbb{R}^n$, estão em posição geral, a célula formada por eles é chamada de Simplexo de dimensão k ou $k-$Simplexo. Denotaremos tal simplexo por $\langle p_0, p_1, ..., p_k \rangle$.



Sub-Simplexos

Dado um simplexo $\sigma = \langle p_0, p_1, ..., p_k \rangle$, cada ponto p_i é chamado de vértice. Os $1-$simplexos gerados pelos pares $[p_i, p_j]$ com $i \neq j$, são chamados de arestas e os $2-$simplexos gerados por $[p_i, p_j, p_k]$ com $i \neq j \neq k$ são chamados de faces de σ.

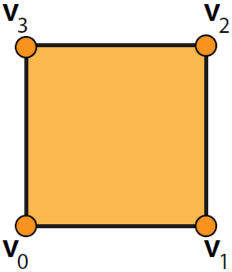


Decomposição celular

Definição

Uma Decomposição Celular de um subconjunto $D \in \mathbb{R}^n$ é um conjunto finito de células $C = \{c_i\}$ que satisfaz as seguintes propriedades:

1. $D = \bigcup_i c_i$;
2. Se $c_i, c_j \in D$, então $c_i \cap c_j \in D$.



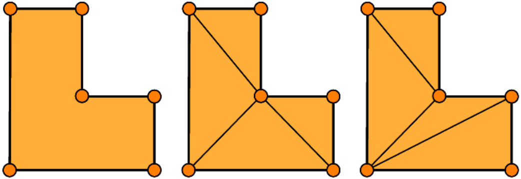
Decomposição celular do quadrado unitário que possui células de 0, 1 e 2 dimensões:

- dimensão 0: v_0, v_1, v_2, v_3
- dimensão 1: $[v_0, v_1], [v_1, v_2], [v_2, v_3], [v_3, v_0]$
- dimensão 2: $[v_0, v_1, v_2, v_3]$

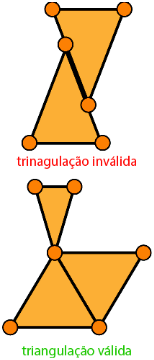
Triangulação

Definição

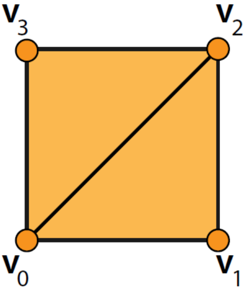
Quando todos os elementos de uma decomposição celular de uma região D são simplexos, dizemos que ela é uma Triangulação de D e denotamos por $\mathcal{T}(D)$.



região *D*　　　duas triangulações de *D*



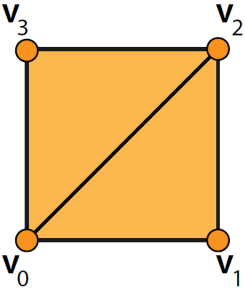
Triangulação de Coxeter-Freudenthal-Kuhn (CFK)



A triangulação do quadrado é formada pelos:

- $0-$Simplexos v_0, v_1, v_2, v_3;
- $1-$Simplexos $\langle v_0, v_1 \rangle, \langle v_1, v_2 \rangle, \langle v_2, v_3 \rangle, \langle v_3, v_0 \rangle$;
- $2-$Simplexos $\langle v_0, v_1, v_2 \rangle, \langle v_0, v_2.v_3 \rangle$.

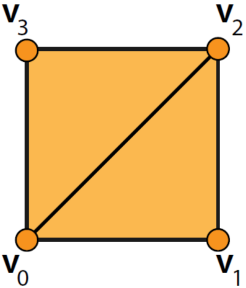
Triangulação de Coxeter-Freudenthal-Kuhn (CFK)



A triangulação do quadrado é formada pelos:

- $0-$Simplexos v_0, v_1, v_2, v_3;
- $1-$Simplexos $\langle v_0, v_1 \rangle, \langle v_1, v_2 \rangle, \langle v_2, v_3 \rangle, \langle v_3, v_0 \rangle$;
- $2-$Simplexos $\langle v_0, v_1, v_2 \rangle, \langle v_0, v_2 . v_3 \rangle$.

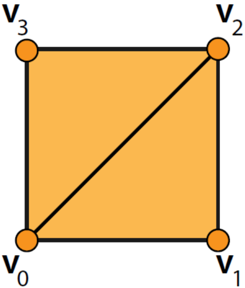
Triangulação de Coxeter-Freudenthal-Kuhn (CFK)



A triangulação do quadrado é formada pelos:

- $0-$Simplexos v_0, v_1, v_2, v_3;
- $1-$Simplexos $\langle v_0, v_1 \rangle, \langle v_1, v_2 \rangle, \langle v_2, v_3 \rangle, \langle v_3, v_0 \rangle$;
- $2-$Simplexos $\langle v_0, v_1, v_2 \rangle, \langle v_0, v_2.v_3 \rangle$.

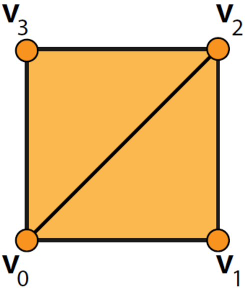
Triangulação de Coxeter-Freudenthal-Kuhn (CFK)



A triangulação do quadrado é formada pelos:

- $0-$Simplexos v_0, v_1, v_2, v_3;
- $1-$Simplexos $\langle v_0, v_1 \rangle, \langle v_1, v_2 \rangle, \langle v_2, v_3 \rangle, \langle v_3, v_0 \rangle$;
- $2-$Simplexos $\langle v_0, v_1, v_2 \rangle, \langle v_0, v_2.v_3 \rangle$.

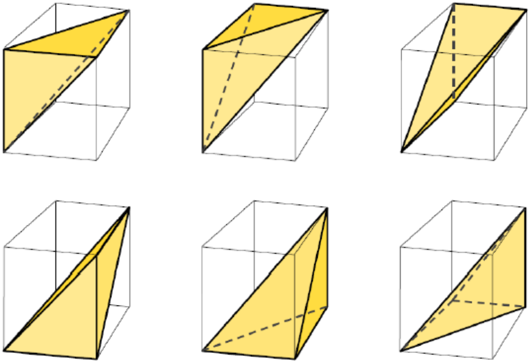
Triangulação de Coxeter-Freudenthal-Kuhn (CFK)



A triangulação do quadrado é formada pelos:

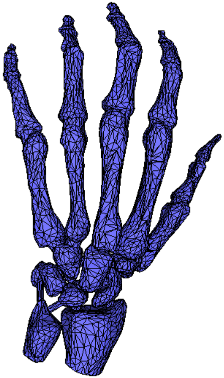
- $0-$Simplexos v_0, v_1, v_2, v_3;
- $1-$Simplexos $\langle v_0, v_1 \rangle, \langle v_1, v_2 \rangle, \langle v_2, v_3 \rangle, \langle v_3, v_0 \rangle$;
- $2-$Simplexos $\langle v_0, v_1, v_2 \rangle, \langle v_0, v_2.v_3 \rangle$.

Triangulação de Coxeter-Freudenthal-Kuhn (CFK)



Usando a diagonal do cubo, podemos decompô-lo em 6 tetraedros (Cada um deles um 3-simplexo).

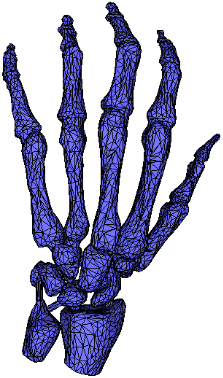
O que armazenar em uma ED?



Geometria (Coordenadas 2D ou 3D);
Atributos do vértice ou da face (Normal, cor, textura, etc...);
Topologia (Relações de Vizinhança ou Conectividade).

Estrutura de Dados

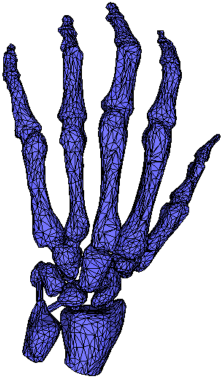
O que armazenar em uma ED?



- Geometria (Coordenadas 2D ou 3D);
- Atributos do vértice ou da face (Normal, cor, textura, etc...);
- Topologia (Relações de Vizinhança ou Conectividade).

Estrutura de Dados

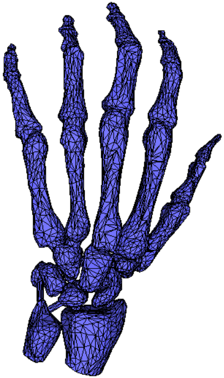
O que armazenar em uma ED?



- Geometria (Coordenadas 2D ou 3D);
- Atributos do vértice ou da face (Normal, cor, textura, etc...);
- Topologia (Relações de Vizinhança ou Conectividade).

Estrutura de Dados

O que armazenar em uma ED?

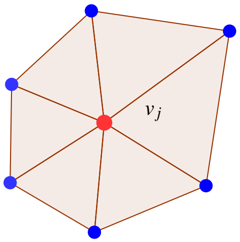


- Geometria (Coordenadas 2D ou 3D);
- Atributos do vértice ou da face (Normal, cor, textura, etc...);
- Topologia (Relações de Vizinhança ou Conectividade).

Estrutura de Dados

O que da ED deve suportar?

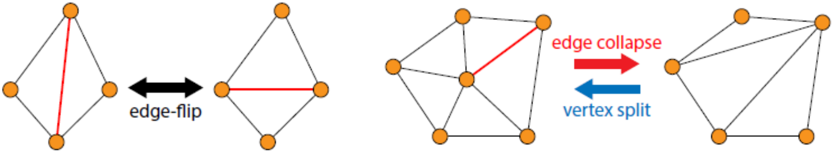
- Rendering;
- Consultas Geométricas;
  - Quais são os vértices de uma determinada face f?
  - Quais são as faces do $1-$anel do vértice v?
  - Quais são as faces adjacentes à face k?



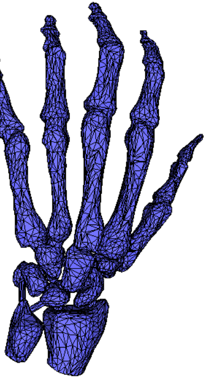
Estrutura de Dados

O que da ED deve suportar?

- Modificações;
  - Remover ou adicionar um vértice / face;
  - *edge-flip*, *edge collpse e vertex split*;



Como avaliar a performance de uma ED?



- Tempo de construção (pré-processamento);
- Tempo de resposta a uma consulta;
- Tempo de realização de uma operação;
- Consumo de memória RAM.

Face Set

- Face: 3 posições;
- Não possui conectividade;
- Arquivos no formato STL;
- Simples, porém redundante.

Triângulos		
(x_1^1, y_1^1, z_1^1)	(x_2^1, y_2^1, z_2^1)	(x_3^1, y_3^1, z_3^1)
(x_1^2, y_1^2, z_1^2)	(x_2^2, y_2^2, z_2^2)	(x_3^2, y_3^2, z_3^2)
(x_1^3, y_1^3, z_1^3)	(x_2^3, y_2^3, z_2^3)	(x_3^3, y_3^3, z_3^3)
$\vdots$	$\vdots$	$\vdots$
(x_1^f, y_1^f, z_1^f)	(x_2^f, y_2^f, z_2^f)	(x_3^f, y_3^f, z_3^f)

Shared Vertex

- Vértice: posição + Face : índice dos vértices;
- Não possui conectividade;
- Arquivos no formato OBJ, OFF, PLY;
- Melhor do que a “Face Set ”,mas ainda com pouca informação.

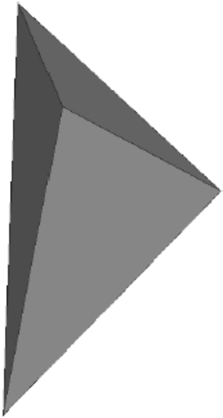
Vértices		
x^1	y^1	z^1
x^2	y^2	z^2
x^3	y^3	z^3
$\vdots$	$\vdots$	$\vdots$
x^v	y^v	z^v

Triângulos		
$\mathbf{v}_1^1$	$\mathbf{v}_2^1$	$\mathbf{v}_3^1$
$\mathbf{v}_1^2$	$\mathbf{v}_2^2$	$\mathbf{v}_3^2$
$\mathbf{v}_1^3$	$\mathbf{v}_2^3$	$\mathbf{v}_3^3$
$\vdots$	$\vdots$	$\vdots$
$\mathbf{v}_1^f$	$\mathbf{v}_2^f$	$\mathbf{v}_3^f$

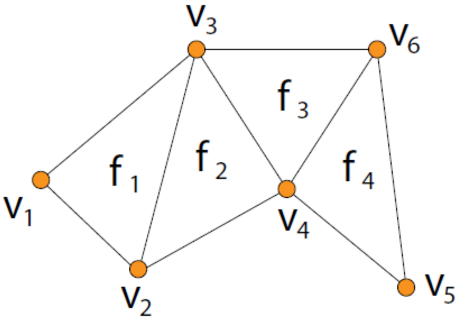
Exemplo: Aquivo OBJ

Tetraedro:

```
# OBJ file format with ext .obj
v 1.0 0.0 0.0
v 0.0 1.0 0.0
v 0.0 0.0 1.0
v 0.0 0.0 0.0
f 2 4 3
f 4 2 1
f 3 1 2
f 1 3 4
```

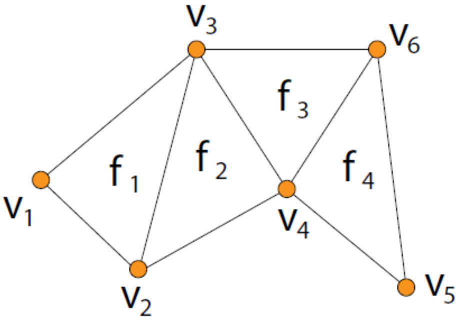


Shared Vertex



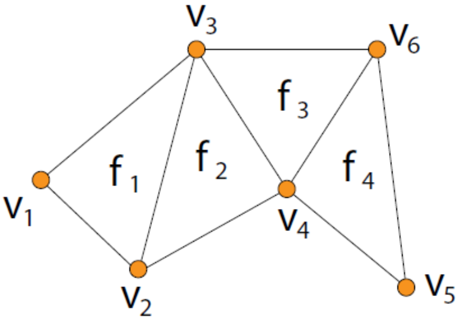
- Quais são os vértices da face f_1?
  - O(1); basta consultar a lista de faces;
- Quais são os vértices so $1-$anel do vértice v_3?
  - Busca completa em todos os vértices;
- Os vértices v_2 e v_6 são adjacentes?

Shared Vertex



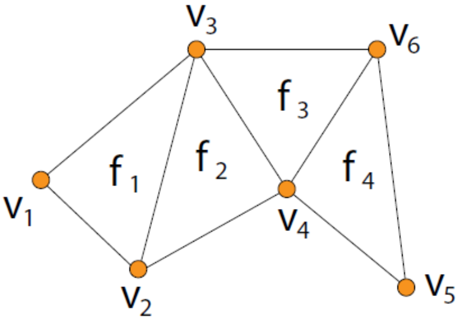
- Quais são os vértices da face f_1?
  - O(1); basta consultar a lista de faces;
- Quais são os vértices so $1-$anel do vértice v_3?
  - Busca completa em todos os vértices;
- Os vértices v_2 e v_6 são adjacentes?

Shared Vertex



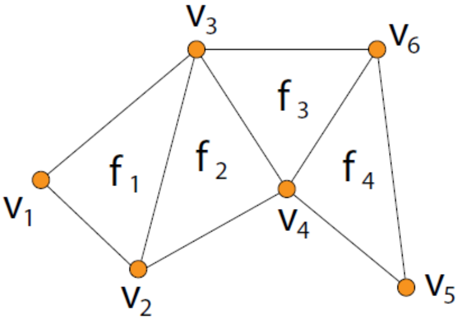
- Quais são os vértices da face f_1?
  - O(1); basta consultar a lista de faces;
- Quais são os vértices so $1-$anel do vértice v_3?
  - Busca completa em todos os vértices;
- Os vértices v_2 e v_6 são adjacentes?
  - Busca completa em todas as faces;

Shared Vertex



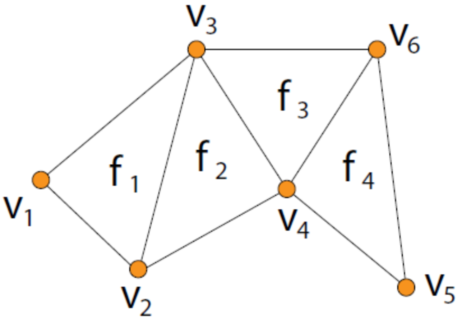
- Quais são os vértices da face f_1?
  - O(1); basta consultar a lista de faces;
- Quais são os vértices so $1-$anel do vértice v_3?
  - Busca completa em todos os vértices;
- Os vértices v_2 e v_6 são adjacentes?
  - Busca completa em todos as faces;

Shared Vertex



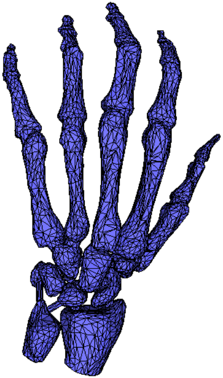
- Quais são os vértices da face f_1?
  - O(1); basta consultar a lista de faces;
- Quais são os vértices so $1-$anel do vértice v_3?
  - Busca completa em todos os vértices;
- Os vértices v_2 e v_6 são adjacentes?
  - Busca completa em todos as faces;

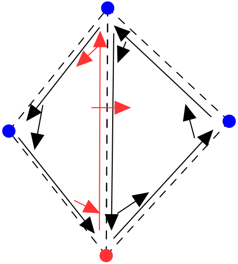
Shared Vertex



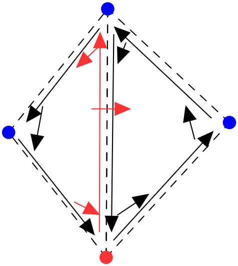
- Quais são os vértices da face f_1?
  - O(1); basta consultar a lista de faces;
- Quais são os vértices so $1-$anel do vértice v_3?
  - Busca completa em todos os vértices;
- Os vértices v_2 e v_6 são adjacentes?
  - Busca completa em todos as faces;

Half Edge



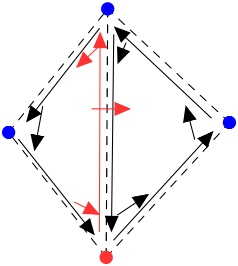


- Vértice
  - Posição
  - 1 HE que “sai ”do vértice.
- Half Edge (HE)
- Face



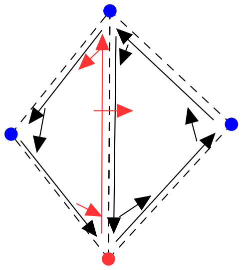
Half Edge

- Vértice
  - Posição
  - 1 HE que “sai ”do vértice.
- Half Edge (HE)
- Face



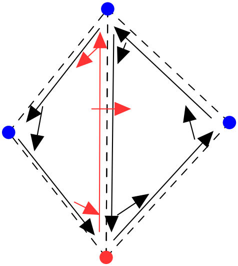
Half Edge

- Vértice
  - Posição
  - 1 HE que “sai ”do vértice.
- Half Edge (HE)
  - Orientação consistente;
  - 1 índice do vértice de origem;
  - 1 índice da face incidente;
  - 1, 2 ou 3 índices de HE's (próxima, anterior e oposta)
- Face



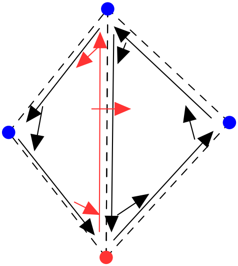
Half Edge

- Vértice
  - Posição
  - 1 HE que “sai ”do vértice.
- Half Edge (HE)
  - Orientação consistente;
  - 1 índice do vértice de origem;
  - 1 índice da face incidente;
  - 1, 2 ou 3 índices de HE's (próxima, anterior e oposta)
- Face



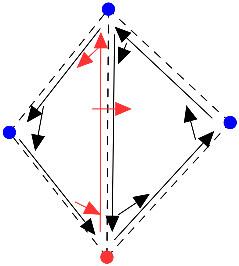
Half Edge

- Vértice
  - Posição
  - 1 HE que “sai ”do vértice.
- Half Edge (HE)
  - Orientação consistente;
  - 1 índice do vértice de origem;
  - 1 índice da face incidente;
  - 1, 2 ou 3 índices de HE’s (próxima, anterior e oposta)
- Face



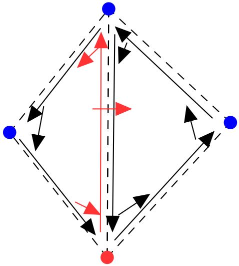
Half Edge

- Vértice
  - Posição
  - 1 HE que “sai ”do vértice.
- Half Edge (HE)
  - Orientação consistente;
  - 1 índice do vértice de origem;
  - 1 índice da face incidente;
  - 1, 2 ou 3 índices de HE’s (próxima, anterior e oposta)
- Face



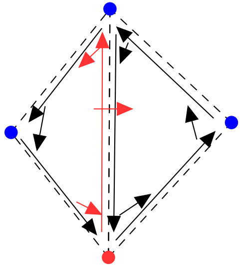
Half Edge

- Vértice
  - Posição
  - 1 HE que “sai ”do vértice.
- Half Edge (HE)
  - Orientação consistente;
  - 1 índice do vértice de origem;
  - 1 índice da face incidente;
  - 1, 2 ou 3 índices de HE’s (próxima, anterior e oposta)
- Face
  - 1 índice de HE incidente.



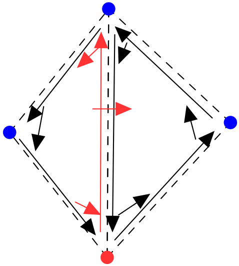
Half Edge

- Vértice
  - Posição
  - 1 HE que “sai ”do vértice.
- Half Edge (HE)
  - Orientação consistente;
  - 1 índice do vértice de origem;
  - 1 índice da face incidente;
  - 1, 2 ou 3 índices de HE’s (próxima, anterior e oposta)
- Face
  - 1 índice de HE incidente.

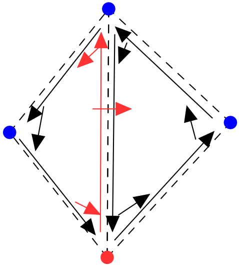


Half Edge

- Vértice
  - Posição
  - 1 HE que “sai ”do vértice.
- Half Edge (HE)
  - Orientação consistente;
  - 1 índice do vértice de origem;
  - 1 índice da face incidente;
  - 1, 2 ou 3 índices de HE’s (próxima, anterior e oposta)
- Face
  - 1 índice de HE incidente.

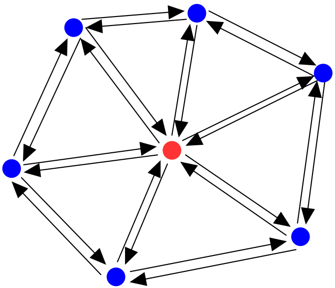


- Vértice
  - Posição
  - 1 HE que “sai ”do vértice.
- Half Edge (HE)
  - Orientação consistente;
  - 1 índice do vértice de origem;
  - 1 índice da face incidente;
  - 1, 2 ou 3 índices de HE’s (próxima, anterior e oposta)
- Face
  - 1 índice de HE incidente.



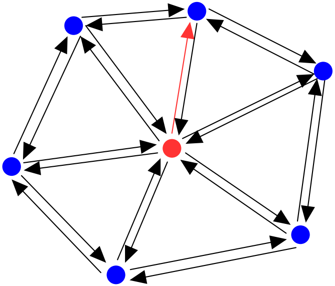
1−Anel com a Half Edge

1. Inicia em um vértice



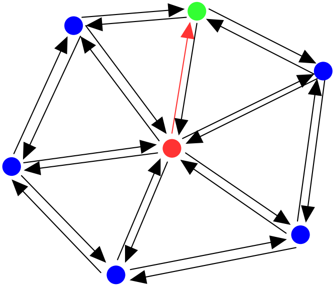
$1-$Anel com a Half Edge

1. Inicia em um vértice
2. HE que sai do vértice



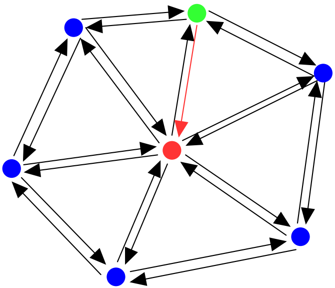
1−Anel com a Half Edge

1. Inicia em um vértice
2. HE que sai do vértice



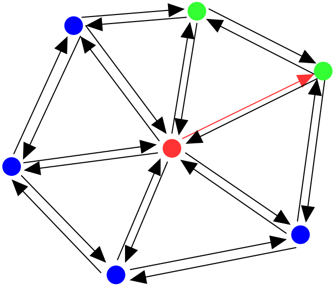
1−Anel com a Half Edge

1. Inicia em um vértice
2. HE que sai do vértice
3. HE oposta



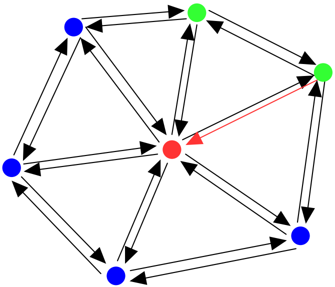
1–Anel com a Half Edge

1. Inicia em um vértice
2. HE que sai do vértice
3. HE oposta
4. Próxima HE



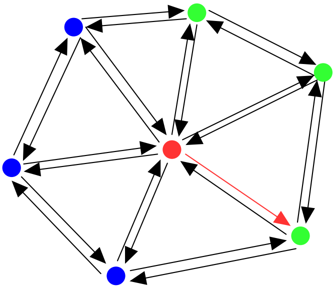
1−Anel com a Half Edge

1. Inicia em um vértice
2. HE que sai do vértice
3. HE oposta
4. Próxima HE
5. HE oposta



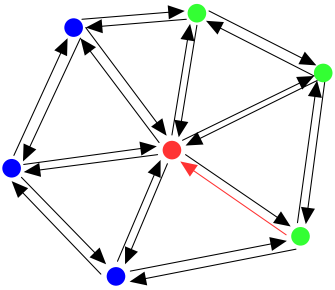
1−Anel com a Half Edge

1. Inicia em um vértice
2. HE que sai do vértice
3. HE oposta
4. Próxima HE
5. HE oposta
6. Próxima HE



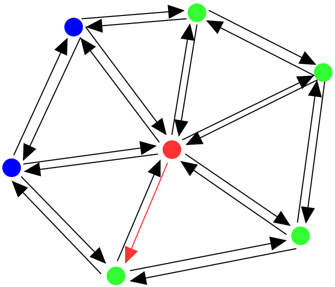
1–Anel com a Half Edge

1. Inicia em um vértice
2. HE que sai do vértice
3. HE oposta
4. Próxima HE
5. HE oposta
6. Próxima HE
7. ...



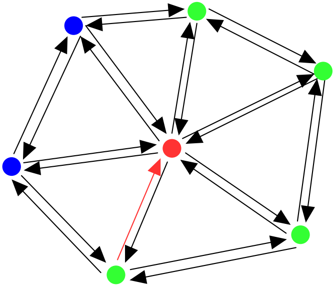
1−Anel com a Half Edge

1. Inicia em um vértice
2. HE que sai do vértice
3. HE oposta
4. Próxima HE
5. HE oposta
6. Próxima HE
7. ...



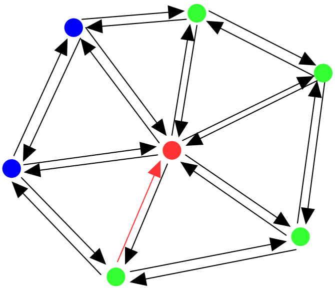
1−Anel com a Half Edge

1. Inicia em um vértice
2. HE que sai do vértice
3. HE oposta
4. Próxima HE
5. HE oposta
6. Próxima HE
7. ...



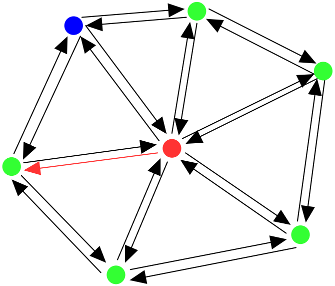
1 – Anel com a Half Edge

1. Inicia em um vértice
2. HE que sai do vértice
3. HE oposta
4. Próxima HE
5. HE oposta
6. Próxima HE
7. ...



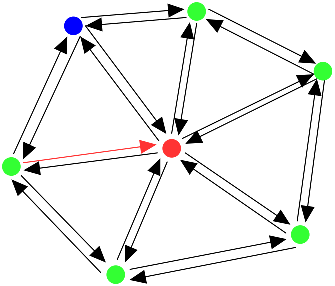
1−Anel com a Half Edge

1. Inicia em um vértice
2. HE que sai do vértice
3. HE oposta
4. Próxima HE
5. HE oposta
6. Próxima HE
7. ...



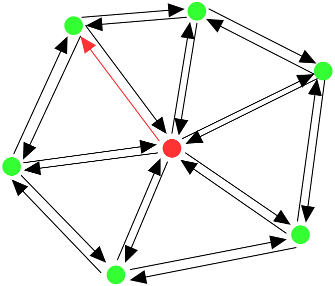
1−Anel com a Half Edge

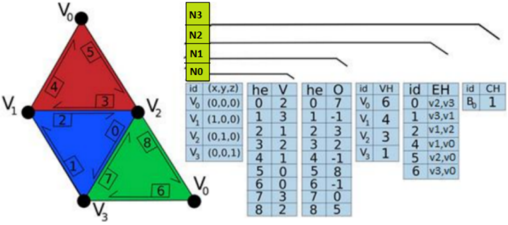
1. Inicia em um vértice
2. HE que sai do vértice
3. HE oposta
4. Próxima HE
5. HE oposta
6. Próxima HE
7. ...



1−Anel com a Half Edge

1. Inicia em um vértice
2. HE que sai do vértice
3. HE oposta
4. Próxima HE
5. HE oposta
6. Próxima HE
7. ...





id	(x,y,z)
V_0	(0,0,0)
V_1	(1,0,0)
V_2	(0,1,0)
V_3	(0,0,1)

he	V
0	2
1	3
2	1
3	2
4	1
5	0
6	0
7	3
8	2

he	O
0	7
1	-1
2	3
3	2
4	-1
5	8
6	-1
7	0
8	5

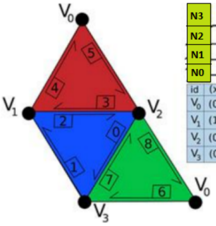
id	VH
V_0	6
V_1	4
V_2	3
V_3	1

id	EH
0	v2,v3
1	v3,v1
2	v1,v2
4	v1,v0
5	v2,v0
6	v3,v0

id	CH
B_0	1

- Dado um triangulo t, sua primeira *halfedge* está na posição $3t$;
- A posição da próxima he é dada por $next_{he} = 3 * \lfloor he/3 \rfloor + (he+1)\%3$;
- A posição he anterior é dada por $prev_{he} = 3 * \lfloor he/3 \rfloor + (he+2)\%3$;

Compact Half Edge [Lages *et al.* 2010]



N3
N2
N1
N0

id	(x,y,z)
V_0	(0,0,0)
V_1	(1,0,0)
V_2	(0,1,0)
V_3	(0,0,1)

he	V
0	2
1	3
2	1
3	2
4	1
5	0
6	0
7	3
8	2

he	O
0	7
1	-1
2	3
3	2
4	-1
5	8
6	-1
7	0
8	5

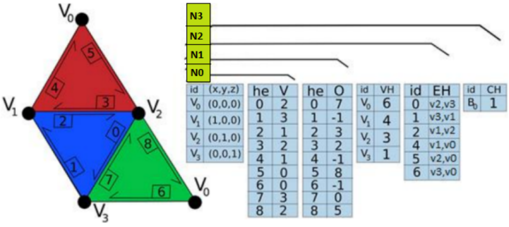
id	VH
V_0	6
V_1	4
V_2	3
V_3	1

id	EH
0	v2,v3
1	v3,v1
2	v1,v2
4	v1,v0
5	v2,v0
6	v3,v0

id	CH
B_0	1

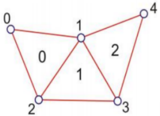
- Dado um triangulo t, sua primeira *halfedge* está na posição $3t$;
- A posição da próxima he é dada por $next_{he} = 3 * \lfloor he/3 \rfloor + (he+1)\%3$;
- A posição he anterior é dada por $prev_{he} = 3 * \lfloor he/3 \rfloor + (he+2)\%3$;

Compact Half Edge [Lages *et al.* 2010]



- Dado um triangulo t, sua primeira *halfedge* está na posição $3t$;
- A posição da próxima he é dada por $next_{he} = 3 * \lfloor he/3 \rfloor + (he+1)\%3$;
- A posição he anterior é dada por $prev_{he} = 3 * \lfloor he/3 \rfloor + (he+2)\%3$;

Opposite Face [Lizier, *et al.* 2006]

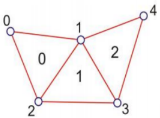


Representação dos vetores da malha

id	Vértices: Coords			Cell
0	X_0	Y_0	Z_0	0
1	X_1	Y_1	Z_1	1
2	X_2	Y_2	Z_2	2
3	X_3	Y_3	Z_3	2
4	X_4	Y_4	Z_4	2

id	Células: V_0	V_1	V_2	Vizinhas		
0	0	1	2	1	-1	-1
1	2	1	3	-1	2	0
2	3	1	4	-1	-1	1

- Vértice:
  - Coordenadas + uma face incidente
- Faces:
  - três índices de vértices + três índices de faces opostas



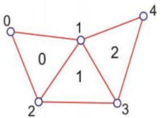
Representação dos vetores da malha

id	Vértices			
	Coords			Cell
0	X_0	Y_0	Z_0	0
1	X_1	Y_1	Z_1	1
2	X_2	Y_2	Z_2	2
3	X_3	Y_3	Z_3	2
4	X_4	Y_4	Z_4	2

id	Células					
	V_0	V_1	V_2	Vizinhas		
0	0	1	2	1	-1	-1
1	2	1	3	-1	2	0
2	3	1	4	-1	-1	1

- Vértice:
  - Coordenadas + uma face incidente
- Faces:
  - três índices de vértices + três índices de faces opostas;

Opposite Face [Lizier, *et al.* 2006]

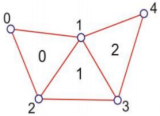


Representação dos vetores da malha

id	Vértices			
	Coords			Cell
0	X_0	Y_0	Z_0	0
1	X_1	Y_1	Z_1	1
2	X_2	Y_2	Z_2	2
3	X_3	Y_3	Z_3	2
4	X_4	Y_4	Z_4	2

id	Células					
	V_0	V_1	V_2	Vizinhas		
0	0	1	2	1	-1	-1
1	2	1	3	-1	2	0
2	3	1	4	-1	-1	1

- Vértice:
  - Coordenadas + uma face incidente
- Faces:
  - três índices de vértices + três índices de faces opostas;



Representação dos vetores da malha

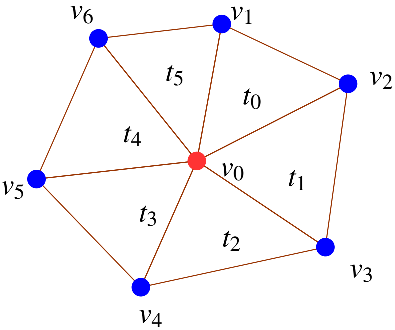
	Vértices			
id	Coords			Cell
0	X_0	Y_0	Z_0	0
1	X_1	Y_1	Z_1	1
2	X_2	Y_2	Z_2	2
3	X_3	Y_3	Z_3	2
4	X_4	Y_4	Z_4	2

	Células					
id	V_0	V_1	V_2	Vizinhas		
0	0	1	2	1	-1	-1
1	2	1	3	-1	2	0
2	3	1	4	-1	-1	1

- Vértice:
  - Coordenadas + uma face incidente
- Faces:
  - três índices de vértices + três índices de faces opostas;

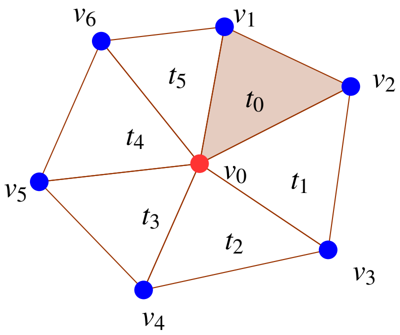
$1-$Anel com a Opposite Face

1. Inicia em um vértice v_0



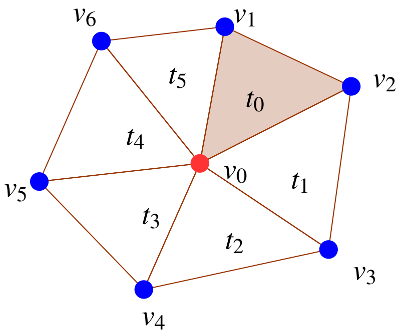
$1-$Anel com a Opposite Face

1. Inicia em um vértice
2. Seja t, uma face incidente em v_0



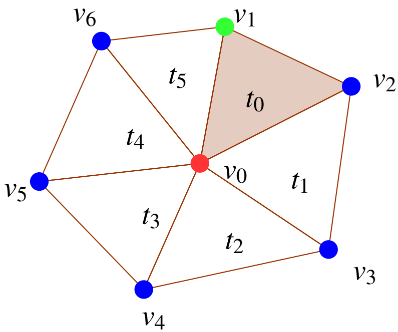
$1-$Anel com a Opposite Face

1. Inicia em um vértice
2. Seja t, uma face incidente em v_0
3. Encontre v_0 na face t



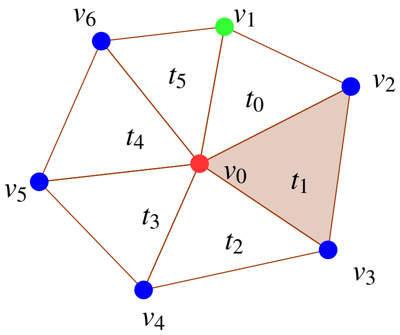
$1-$Anel com a Opposite Face

1. Inicia em um vértice
2. Seja t, uma face incidente em v_0
3. Encontre v_0 na face t
4. Próximo vértice a partir de v_0



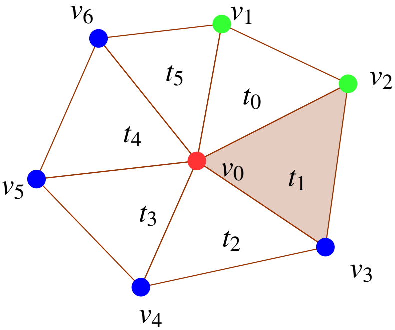
$1-$Anel com a Opposite Face

1. Inicia em um vértice
2. Seja t, uma face incidente em v_0
3. Encontre v_0 na face t
4. Próximo vértice a partir de v_0
5. Seja t a face oposta a v_1



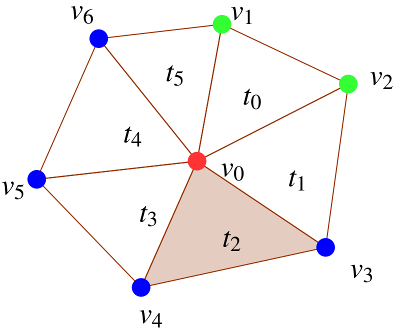
$1-$Anel com a Opposite Face

1. Inicia em um vértice
2. Seja t, uma face incidente em v_0
3. Encontre v_0 na face t
4. Próximo vértice a partir de v_0
5. Seja t a face oposta a v_1
6. Voltar ao passo 3 ...



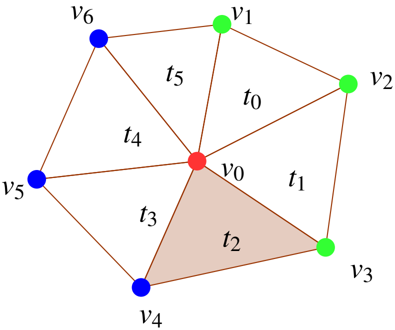
$1-$Anel com a Opposite Face

1. Inicia em um vértice
2. Seja t, uma face incidente em v_0
3. Encontre v_0 na face t
4. Próximo vértice a partir de v_0
5. Seja t a face oposta a v_1
6. Voltar ao passo 3 ...



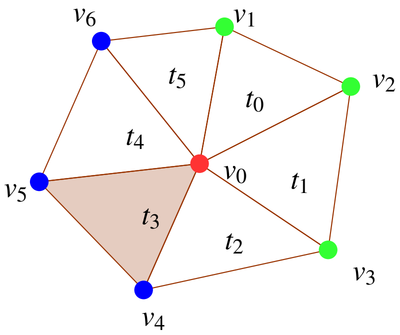
1−Anel com a Opposite Face

1. Inicia em um vértice
2. Seja t, uma face incidente em v_0
3. Encontre v_0 na face t
4. Próximo vértice a partir de v_0
5. Seja t a face oposta a v_1
6. Voltar ao passo 3 ...



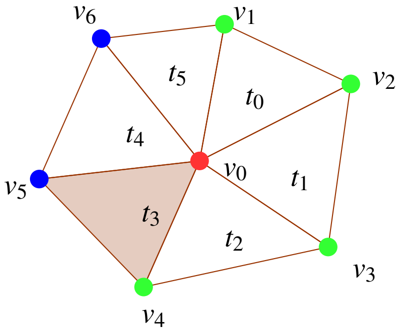
$1-$Anel com a Opposite Face

1. Inicia em um vértice
2. Seja t, uma face incidente em v_0
3. Encontre v_0 na face t
4. Próximo vértice a partir de v_0
5. Seja t a face oposta a v_1
6. Voltar ao passo 3 ...



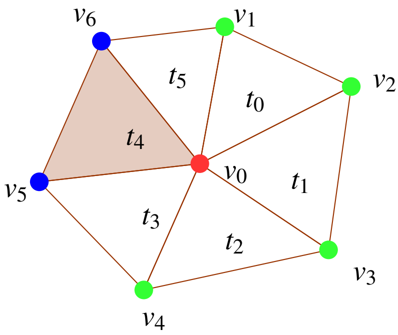
$1-$Anel com a Opposite Face

1. Inicia em um vértice
2. Seja t, uma face incidente em v_0
3. Encontre v_0 na face t
4. Próximo vértice a partir de v_0
5. Seja t a face oposta a v_1
6. Voltar ao passo 3 ...



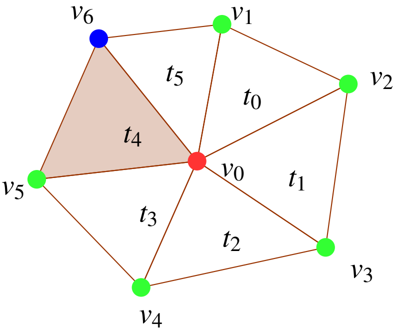
$1-$Anel com a Opposite Face

1. Inicia em um vértice
2. Seja t, uma face incidente em v_0
3. Encontre v_0 na face t
4. Próximo vértice a partir de v_0
5. Seja t a face oposta a v_1
6. Voltar ao passo 3 ...



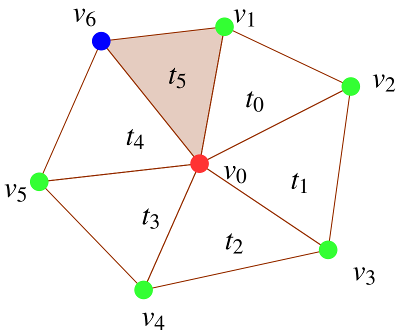
$1-$Anel com a Opposite Face

1. Inicia em um vértice
2. Seja t, uma face incidente em v_0
3. Encontre v_0 na face t
4. Próximo vértice a partir de v_0
5. Seja t a face oposta a v_1
6. Voltar ao passo 3 ...



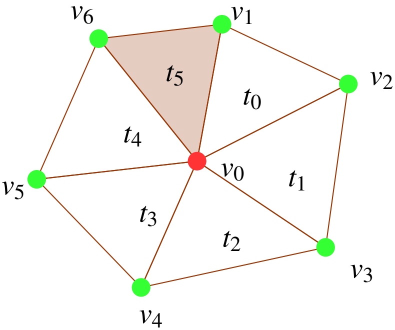
$1-$Anel com a Opposite Face

1. Inicia em um vértice
2. Seja t, uma face incidente em v_0
3. Encontre v_0 na face t
4. Próximo vértice a partir de v_0
5. Seja t a face oposta a v_1
6. Voltar ao passo 3 ...

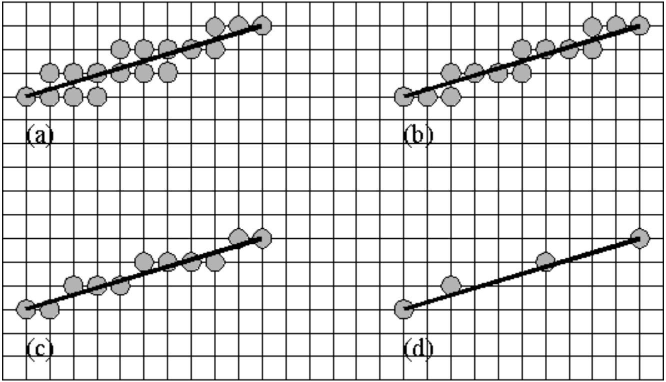


$1-$Anel com a Opposite Face

1. Inicia em um vértice
2. Seja t, uma face incidente em v_0
3. Encontre v_0 na face t
4. Próximo vértice a partir de v_0
5. Seja t a face oposta a v_1
6. Voltar ao passo 3 ...



Traçado de “Curvas ”:



Traçado de “Curvas ”:

Características Desejáveis

1. Linearidade: os pixels traçados devem dar a aparência de que estão sobre uma reta. Isto é trivial no caso de segmentos paralelos aos eixos x ou y, ou com inclinação de 45 graus, mas não nos outros casos;
2. Precisão: Os segmentos devem terminar e iniciar nos pontos especificados;
3. Espessura (Densidade) uniforme: a densidade da linha é dada pelo número de pixels traçados dividido pelo comprimento da linha;
4. Intensidade independente da inclinação: para segmentos de diferentes inclinações;
5. Continuidade: a imagem não deve apresentar interrupções indesejáveis;
6. Rapidez:

Traçado de “Curvas ”:

Algoritmo DDA (Digital Differential analyzer)

Dada a Expressão $y = \frac{\Delta y}{\Delta x}x + b$; obter o incremento das coordenadas (x_p, y_p) de modo a obter o ponto subsequente (x_{p+1}, y_{p+1}).

1. em função do diferencial $\frac{\Delta y}{\Delta x}$;
2. Para $m \leq 1$, as coordenadas x crescem mais rapidamente que as coordenadas y. Portanto, a amostragem é feita incrementando unitariamente na direção x. E na direção y:

$$y_{p+1} = y_p + m$$

3. Se $m \geq 1$ faz-se incremento unitário na direção y. E na direção x:

$$x_{p+1} = x_p + \frac{1}{m}$$

Traçado de "Curvas ":

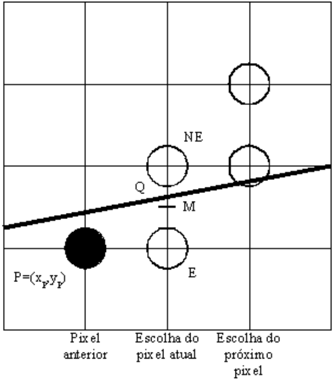
Algoritmo Do Ponto médio, ou de Bresenham.

Características.

1. Não executa operações em ponto flutuante;
2. Assume que a inclinação m está entre 0 e 1. Para outros valores podem ser tratados por simetria;
3. O ponto inicial $P_i = (x_0, y_0)$ está no canto inferior esquerdo e o ponto final $P_f = (x_n, y_n)$ no canto superior direito.

Traçado de “Curvas ”:

Algoritmo Do Ponto médio, ou de Bresenham.



Traçado de “Curvas ”:

Algoritmo Do Ponto médio, ou de Bresenham.

- Precisamos de um método para avaliar de que lado da reta está o ponto M;
- Partimos da equaçao da reta $y = \frac{\Delta y}{\Delta x} x + B$;
- Derivamos a equação implícita da mesma: Tomando que $\Delta y = y_2 - y_1$ e $\Delta x = x_2 - x_1$:
- Teremos $F(x,y) = \Delta y.x - \Delta x.y + B.\Delta x = 0$.

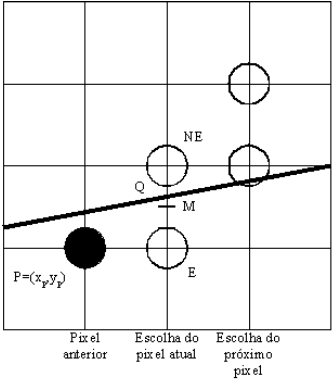
Traçado de “Curvas ”:

Algoritmo Do Ponto médio, ou de Bresenham.

- Sabemos que $F(x,y) = 0$ sobre a reta, positiva abaixo dela e negativa acima;
- Basta calcular $F(M) = F(x_p + 1, y_p + \frac{1}{2})$;
- Veja que somente é necessário saber o sinal de $F(M)$

Traçado de “Curvas ”:

Algoritmo Do Ponto médio, ou de Bresenham.



Traçado de “Curvas ”:

Algoritmo Do Ponto médio, ou de Bresenham.

- Definimos uma variável de decisão $d_p = a(x_p + 1) + b(y_p + \frac{1}{2}) + c$;
- Se $d_p > 0$ escolhemos o pixel NE; se for menor, o pixel E será escolhido; e se for nulo qualquer um dos dois pode ser escolhido;
- Observe que a variável de decisão pode ser calculada de forma recorrente na interação $p + 1$;

Traçado de “Curvas ”:

Algoritmo Do Ponto médio, ou de Bresenham.

- Se $d_p \geq 0$ segue que $y_{p+1} = y_p$;

$$
\begin{aligned}
d_{p+1} = F\left(x_p + 2, y_{p+1} + \frac{1}{2}\right) &= \\
a\,(x_p + 2) + b\left(y_p + \frac{1}{2}\right) + c &= \\
a(x_p + 1) + a + b(y_p + \frac{1}{2}) + c &= \\
F\,(x_p + 1, y_p + \frac{1}{2}) + a &= \\
d_p + a & .
\end{aligned}
$$

Traçado de "Curvas ":

Algoritmo Do Ponto médio, ou de Bresenham.

- Se $d_p < 0$ segue que $y_{p+1} = y_p + 1$;

$$\begin{aligned}
d_{p+1} = F\left(x_p + 2, y_{p+1} + \frac{1}{2}\right) &= \\
a\left(x_p + 2\right) + b\left(y_p + 1\frac{1}{2}\right) + c &= \\
a(x_p + 1) + a + b(y_p + \frac{1}{2}) + b + c &= \\
F\left(x_p + 1, y_p + \frac{1}{2}\right) + a + b &= \\
d_p + a + b & .
\end{aligned}$$

Traçado de “Curvas ”:

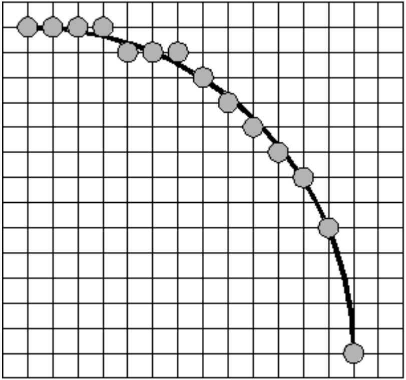
Algoritmo Do Ponto médio, ou de Bresenham.

- O valor de d_0 pode ser obtido diretamente a partir da expressão:

$$
\begin{aligned}
d_0 = F(x_0+1, y_0+\frac{1}{2}) &= \\
a(x_0+1) + b\left(y_0+\frac{1}{2}\right) + c &= \\
a(x_0) + a + b(y_0) + \frac{1}{2}b + c &= \\
F(x_0, y_0) + a + \frac{1}{2}b & .
\end{aligned}
$$

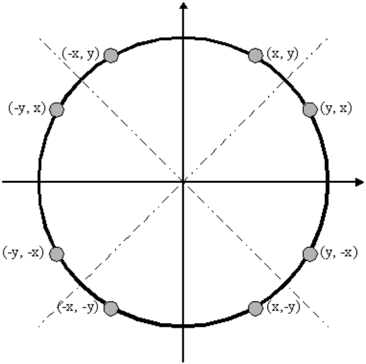
Traçado de “Curvas ”:

Algoritmo Do Ponto médio, ou de Bresenham para circunferências.



Traçado de "Curvas ":

Algoritmo Do Ponto médio, ou de Bresenham para circunferências.
Aproveitamento da Simetria - Somente é necessário calcular o segundo octante:



Traçado de “Curvas ”:

Algoritmo Do Ponto médio, ou de Bresenham para circunferências.
Aproveitamento da Simetria - Somente é necessário calcular o segundo octante:

$$(x-a)^2+(y-b)^2=R^2$$

onde:

- O ponto (a,b) é o centro da circunferência;
- O valor R representa o seu raio.

Traçado de “Curvas ”:

Algoritmo Do Ponto médio, ou de Bresenham para circunferências.
Aproveitamento da Simetria :

$$F(x,y) = (x-a)^2 + (y-b)^2 - R^2 = 0$$

onde:

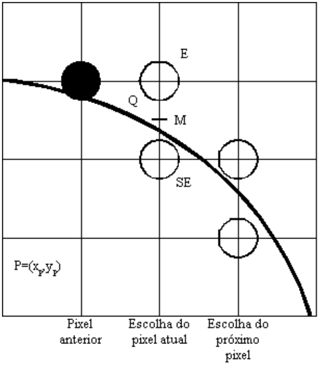
- A função $F(x,y)$ é zero sobre a circunferência;
- É negativa dentro dela e positiva fora.

Traçado de “Curvas ”:

Algoritmo Do Ponto médio, ou de Bresenham para circunferências.

Aproveitamento da Simetria :

Dado o ponto $P = (x_p, y_p)$, escolher entre os pontos E e SE.



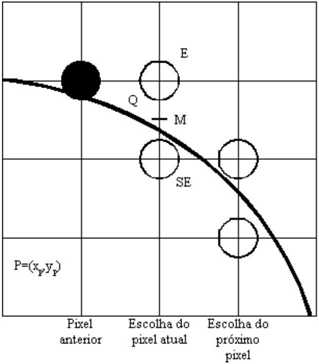
Traçado de “Curvas ”:

Algoritmo Do Ponto médio, ou de Bresenham para circunferências.

Aproveitamento da Simetria :

A variável de decisão no ponto $P = (x_p, y_p)$ é calculada por

$$d_p = F(x_p + 1, y_p - \frac{1}{2}) = ((x_p + 1) - a)^2 + ((y_p - \frac{1}{2}) - b)^2 - R^2$$



Traçado de “Curvas ”:

Algoritmo Do Ponto médio, ou de Bresenham para circunferências.

- Se $d_p \geq 0$ segue que $y_{p+1} = y_p - 1$;

$$d_p = (x_p - a)^2 + 2\,(x_p - a) + (y_p - b)^2 + (b - y_p) + \frac{5}{4} - R^2$$

$$\begin{aligned} d_{p+1} = F\left(x_p + 2, y_p - \frac{3}{2}\right) &= \\ (x_p - a)^2 + 4\,(x_p - a) + (y_p - b)^2 + 3\,(b - y_p) + \frac{25}{4} - R^2 &= \\ d_p + 2(x_p - a) - 2(y_p - b) + 5 &. \end{aligned}$$

Traçado de “Curvas ”:

Algoritmo Do Ponto médio, ou de Bresenham para circunferências.

- Se $d_p < 0$ segue que $y_{p+1} = y_p$;

$$d_p = (x_p - a)^2 + 2\,(x_p - a) + (y_p - b)^2 + (b - y_p) + \frac{5}{4} - R^2$$

$$\begin{aligned} d_{p+1} = F\left(x_p + 2, y_p - \frac{1}{2}\right) &= \\ (x_p - a)^2 + 4\,(x_p - a) + (y_p - b)^2 + (b - y_p) + \frac{17}{4} - R^2 &= \\ d_p + 2(x_p - a) + 3 &. \end{aligned}$$

Traçado de “Curvas ”:

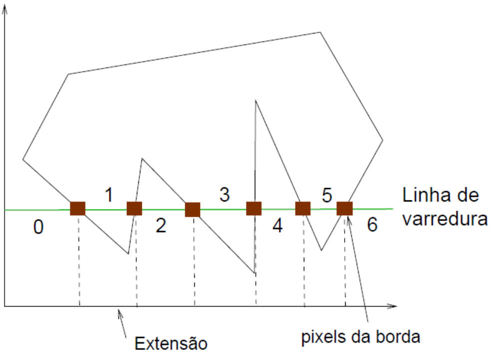
Algoritmo Do Ponto médio, ou de Bresenham para circunferências.

Exercício:

Encontre uma expressão para d_0:

Preenchimento:

Preenchimento de Polígonos.



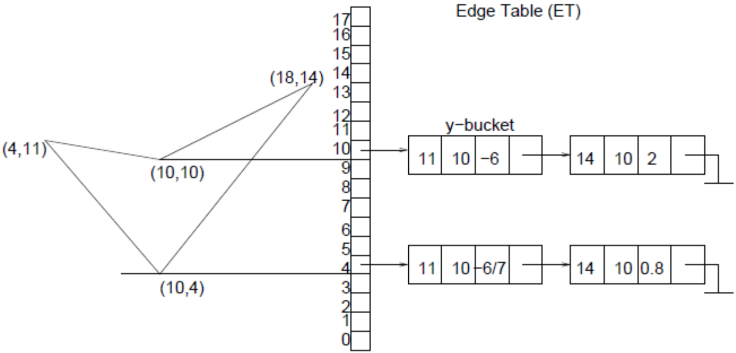
Preenchimento:

Preenchimento de Polígonos.
Se baseia nos seguinte fatos:

1. Os pixels das arestas de borda podem ser obtidos com o auxílio do algoritmo de traçado de retas;
2. Para cada linha de varredura, os conjuntos de pixels são divididos em diferentes extensões (extent) separados por pixels correspondentes as arestas de bordo;
3. A classificação de pertinência das das extensões identificadas em cada linha se alterna ao longo de uma linha de varredura.

Preenchimento:

Preenchimento de Polígonos.
Edge table



Preenchimento:

Preenchimento de Polígonos.

y - buckets

1. Cada y-bucket associado a uma linha contém informações de uma aresta do polígono que a intercepta;
2. São Armazenados:
    1. Coordenada y do ponto extremo que tem maior $y \rightarrow (y_{max})$;
    2. Coordenada x do ponto extremo que tem menor $y \rightarrow (x_{min})$;
    3. A inclinação da reta $\frac{\Delta x}{\Delta y} = \frac{1}{m}$.

Preenchimento:

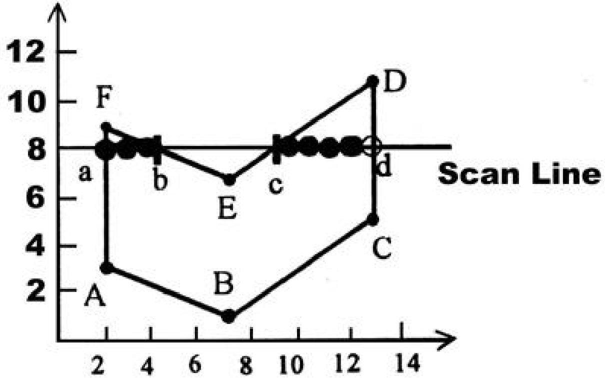
Preenchimento de Polígonos.

y - buckets

- Para cada linha de varredura y, é mantida uma lista de buckets ativos ordenadas de maneira crescente em função do valor do seu campo x_{min};
- O preenchimento da linha é feito de forma alternada entre os intervalos dos segmentos delimitados por estes pontos.

Preenchimento:

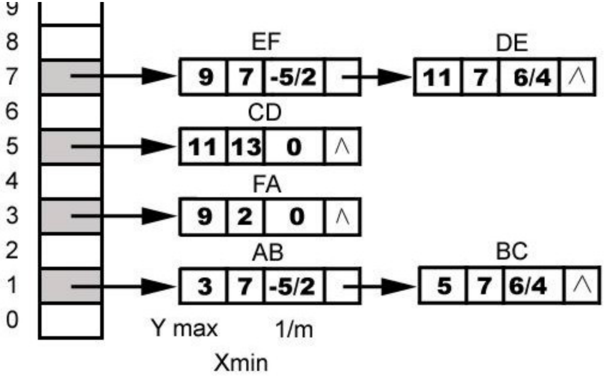
Preenchimento de Polígonos.
Exemplo:



Preenchimento:

Preenchimento de Polígonos.
Exemplo:

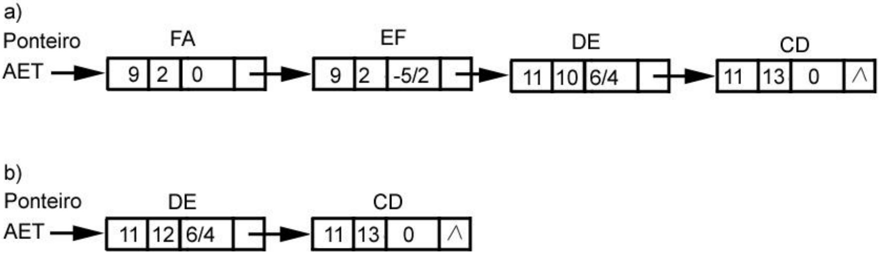
Edge Table



Preenchimento:

Preenchimento de Polígonos.
Exemplo:

Buckets Ativos



Preenchimento:

Obtenha a menor coordenada y armazenada na ET;
Inicialize a AET como vazia;
enquanto *AET e ET não estiverem vazias* **faça**
    Transfira do cesto y na ET para a AET as arestas cujo ymin = y , mantendo a AET ordenada em x;
    Retira os lados que possuem $y = y_{max}$;
    Desenhe os pixels do bloco na linha de varredura y usando pares de coordenadas x da AET;
    Incremente y de 1 ;
    Para cada aresta não vertical que permanece na AET, atualiza x para o novo y;
    Como o passo anterior pode ter desordenado a AET, reordena a AET;
fim

Algoritmo 1: Preenchimento de Polígonos

Preenchimento:

Preenchimento de Polígonos.

OBS:

Podemos integrar o algoritmo de preenchimento de polígonos com os algoritmos de tonalização, calculado a cor do pixel que será preenchido pelo mesmo;

FIM