

---

# Curso de Fotografia 3D: Geração de Malhas com Ball-Pivoting



por Esdras Medeiros e Luiz Velho

# Sumário

---

1. O Problema;

# Sumário

---

1. O Problema;
2. Algoritmos de Avanço de Frente;

# Sumário

---

1. O Problema;
2. Algoritmos de Avanço de Frente;
3. Topologia de Superfícies;

# Sumário

---

1. O Problema;
2. Algoritmos de Avanço de Frente;
3. Topologia de Superfícies;
4. Teoria de *Handlebody*;

# Sumário

---

1. O Problema;
2. Algoritmos de Avanço de Frente;
3. Topologia de Superfícies;
4. Teoria de *Handlebody*;
5. Operadores de *Handle*

# Sumário

---

1. O Problema;
2. Algoritmos de Avanço de Frente;
3. Topologia de Superfícies;
4. Teoria de *Handlebody*;
5. Operadores de *Handle*
6. O Algoritmo;

# Sumário

---

1. O Problema;
2. Algoritmos de Avanço de Frente;
3. Topologia de Superfícies;
4. Teoria de *Handlebody*;
5. Operadores de *Handle*
6. O Algoritmo;
7. Aceleração;



# Sumário

---

1. O Problema;
2. Algoritmos de Avanço de Frente;
3. Topologia de Superfícies;
4. Teoria de *Handlebody*;
5. Operadores de *Handle*
6. O Algoritmo;
7. Aceleração;
8. Conclusão.

# 1. O Problema

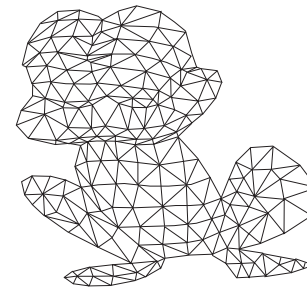
---

Dado um conjunto finito de pontos  $P \subset \mathbb{R}^3$  gerar uma superfície  $S$  tal que  $P \subset S$  ou  $\max \{ \|p - S\| : p \in P \}$  é suficientemente pequeno.

Nuvem de pontos  
tridimensionais



Malha extraída



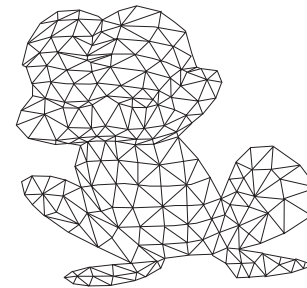
# 1. O Problema

Dado um conjunto finito de pontos  $P \subset \mathbb{R}^3$  gerar uma superfície  $S$  tal que  $P \subset S$  ou  $\max \{ \|p - S\| : p \in P \}$  é suficientemente pequeno.

Nuvem de pontos  
tridimensionais



Malha extraída

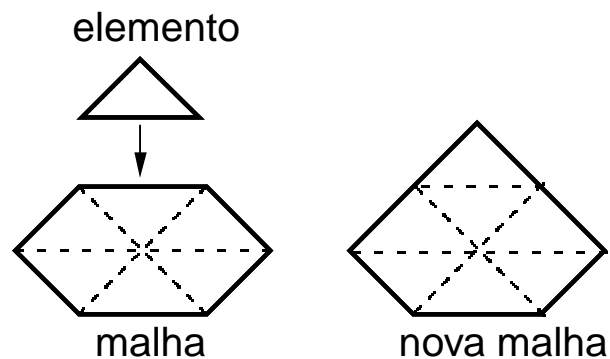


Em geral,  $S$  é uma superfície linear por partes (triangulada), ou seja, uma poligonização comumente denominada por malha.

# 2. Algoritmos de Avanço de Frente (AAF)

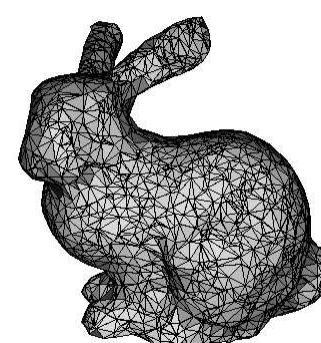
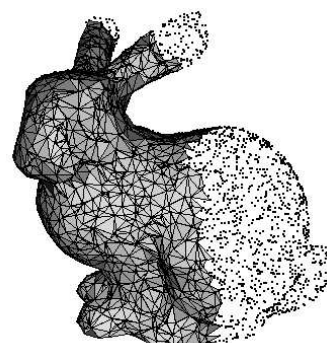
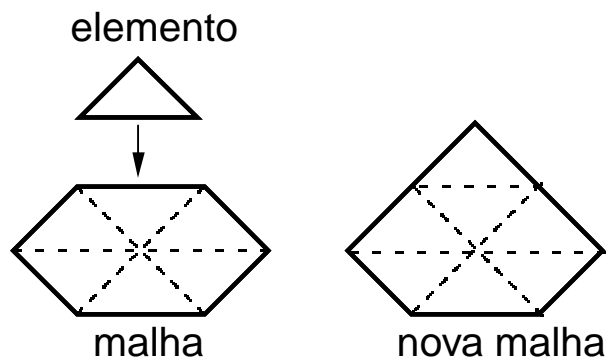
---

A malha é construída progressivamente associando triângulos ao seu bordo a partir de algum critério geométrico.



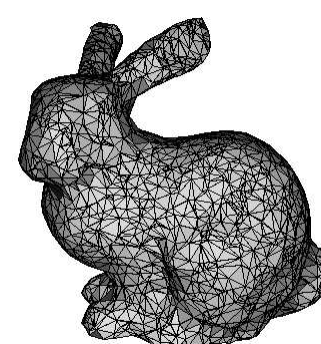
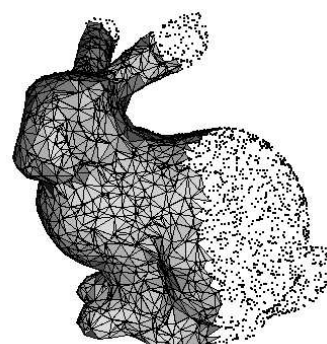
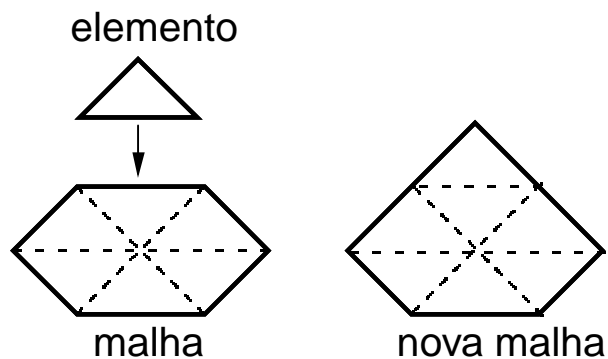
# 2. Algoritmos de Avanço de Frente (AAF)

A malha é construída progressivamente associando triângulos ao seu bordo a partir de algum critério geométrico.



# 2. Algoritmos de Avanço de Frente (AAF)

A malha é construída progressivamente associando triângulos ao seu bordo a partir de algum critério geométrico.



**Dificuldade:** é preciso tratar as diversas mudanças topológicas.

# 3. Topologia de Superfícies

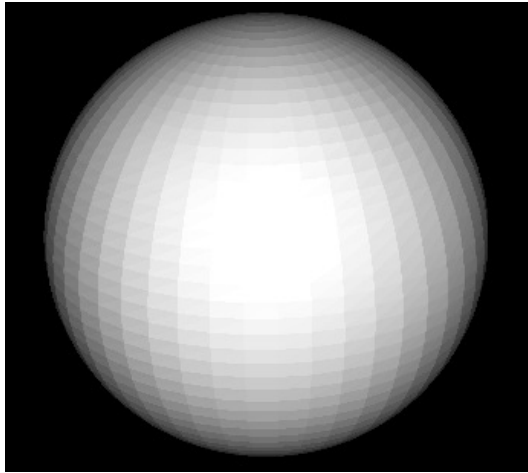
---

Trata de equivalências por deformações:

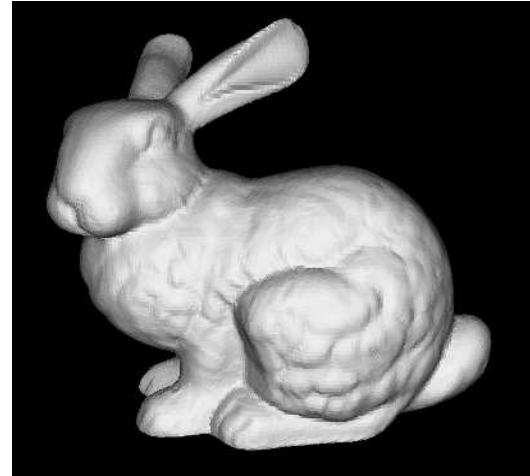
# 3. Topologia de Superfícies

---

Trata de equivalências por deformações:



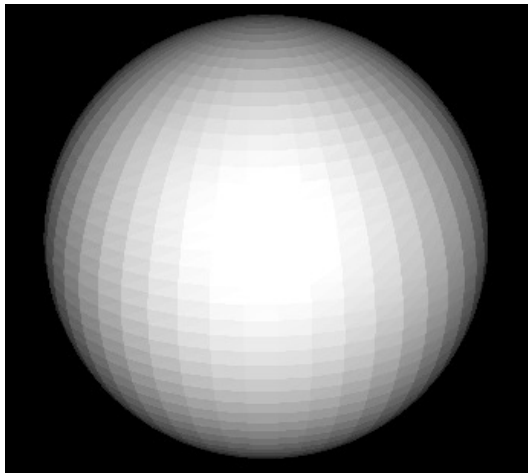
=



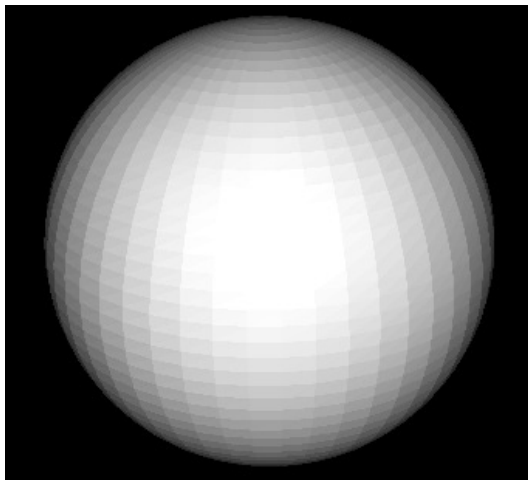
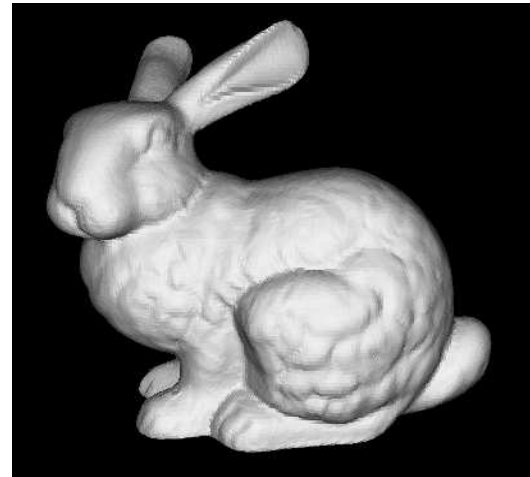


# 3. Topologia de Superfícies

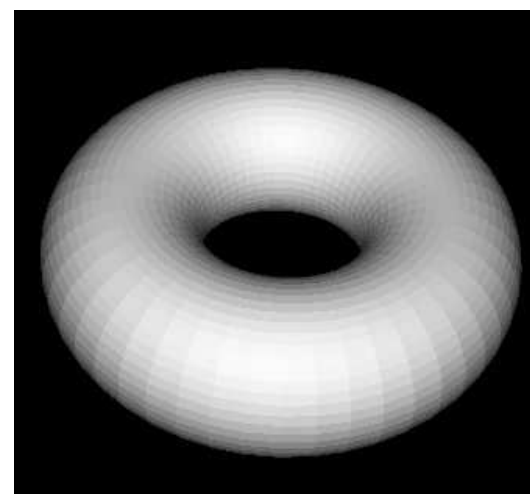
Trata de equivalências por deformações:



=



≠



# 3. Topologia de Superfícies

---

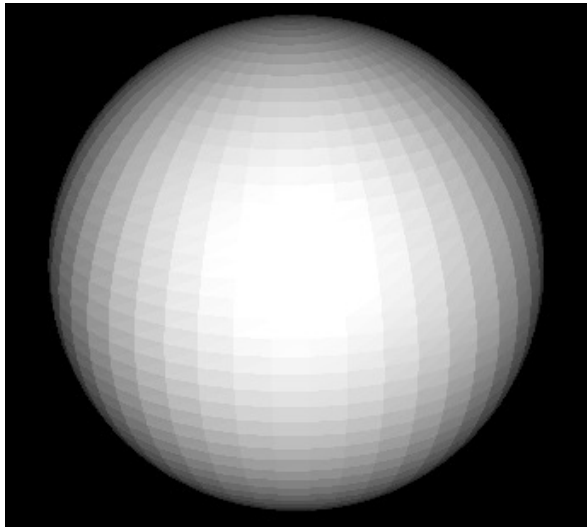
O número de curvas de bordo é uma variante topológica:

Ex:

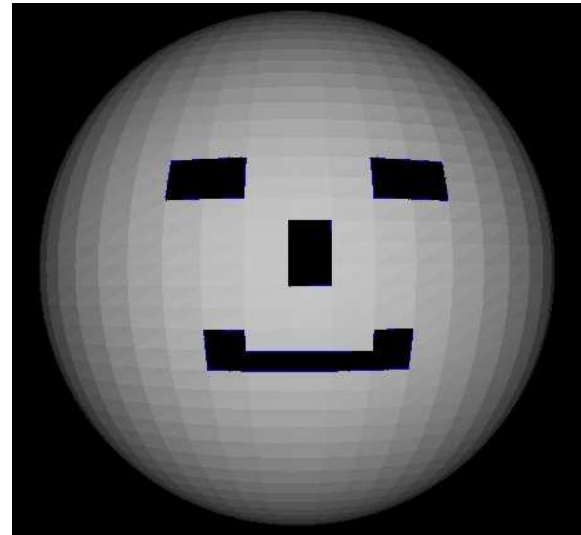
# 3. Topologia de Superfícies

---

O número de curvas de bordo é uma variante topológica:  
Ex:



$\neq$



# 3. Topologia de Superfícies

---

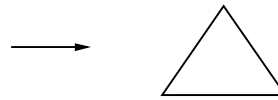
Tipos de mudança da topologia em AAF's:

# 3. Topologia de Superfícies

---

Tipos de mudança da topologia em AAF's:

1. Criação de uma curva:

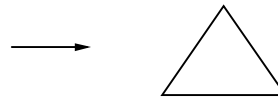


# 3. Topologia de Superfícies

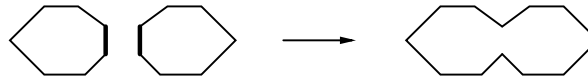
---

Tipos de mudança da topologia em AAF's:

1. Criação de uma curva:



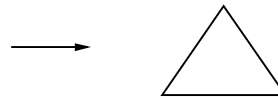
2. Duas curvas se juntam formando uma só:



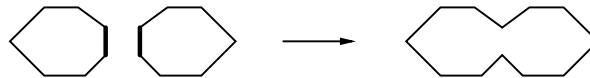
# 3. Topologia de Superfícies

Tipos de mudança da topologia em AAF's:

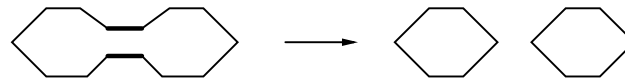
1. Criação de uma curva:



2. Duas curvas se juntam formando uma só:



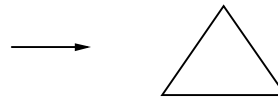
3. Uma curva divide-se em duas:



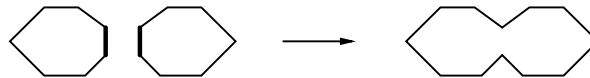
# 3. Topologia de Superfícies

Tipos de mudança da topologia em AAF's:

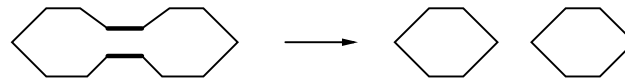
1. Criação de uma curva:



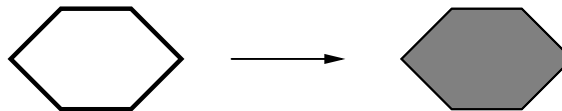
2. Duas curvas se juntam formando uma só:



3. Uma curva divide-se em duas:



4. Uma curva é fechada:

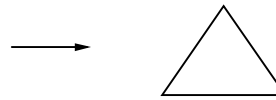




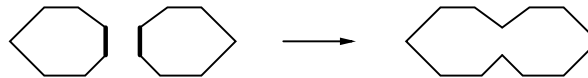
# 3. Topologia de Superfícies

Tipos de mudança da topologia em AAF's:

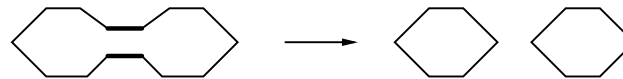
1. Criação de uma curva:



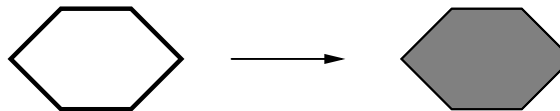
2. Duas curvas se juntam formando uma só:



3. Uma curva divide-se em duas:



4. Uma curva é fechada:



**A teoria de handlebody ferramenta matemática ideal para abordar  
AAF's.**

# 4. Teoria de Handlebody

---

**Definição.**  $H_\lambda = (A_\lambda, B_\lambda)$  é um *handle* de índice  $\lambda = 0, 1, 2$  t.q.  
 $B_\lambda \subseteq \partial A_\lambda$  onde  $A_\lambda = D^\lambda \times D^{2-\lambda}$  e  $B_\lambda = (\partial D^\lambda) \times D^{2-\lambda}$ .

# 4. Teoria de Handlebody

---

**Definição.**  $H_\lambda = (A_\lambda, B_\lambda)$  é um *handle* de índice  $\lambda = 0, 1, 2$  t.q.  
 $B_\lambda \subseteq \partial A_\lambda$  onde  $A_\lambda = D^\lambda \times D^{2-\lambda}$  e  $B_\lambda = (\partial D^\lambda) \times D^{2-\lambda}$ .

Tipo-0,  $\lambda = 0$ :

$$\begin{aligned} A_0 &= D^0 \times D^2 = \bullet \\ B_0 &= (\partial D^0) \times D^2 = \emptyset \end{aligned}$$

# 4. Teoria de Handlebody

**Definição.**  $H_\lambda = (A_\lambda, B_\lambda)$  é um *handle* de índice  $\lambda = 0, 1, 2$  t.q.  
 $B_\lambda \subseteq \partial A_\lambda$  onde  $A_\lambda = D^\lambda \times D^{2-\lambda}$  e  $B_\lambda = (\partial D^\lambda) \times D^{2-\lambda}$ .

Tipo-0,  $\lambda = 0$ :

$$A_0 = D^0 \times D^2 = \text{●}$$
$$B_0 = (\partial D^0) \times D^2 = \emptyset$$

Tipo-1,  $\lambda = 1$ :

$$A_1 = D^1 \times D^1 = \text{■}$$
$$B_1 = (\partial D^1) \times D^1 = \text{! !}$$

# 4. Teoria de Handlebody

**Definição.**  $H_\lambda = (A_\lambda, B_\lambda)$  é um *handle* de índice  $\lambda = 0, 1, 2$  t.q.  $B_\lambda \subseteq \partial A_\lambda$  onde  $A_\lambda = D^\lambda \times D^{2-\lambda}$  e  $B_\lambda = (\partial D^\lambda) \times D^{2-\lambda}$ .

Tipo-0,  $\lambda = 0$ :

$$\begin{aligned} A_0 &= D^0 \times D^2 = \text{●} \\ B_0 &= (\partial D^0) \times D^2 = \emptyset \end{aligned}$$

Tipo-1,  $\lambda = 1$ :

$$\begin{aligned} A_1 &= D^1 \times D^1 = \text{■} \\ B_1 &= (\partial D^1) \times D^1 = \text{! !} \end{aligned}$$

Tipo-2,  $\lambda = 2$ :

$$\begin{aligned} A_2 &= D^2 \times D^0 = \text{●} \\ B_2 &= (\partial D^2) \times D^0 = \text{○} \end{aligned}$$

# 4. Teoria de *Handlebody*

---

**Teorema.** (Decomposição de Superfícies por *Handlebodies*)  
Para toda superfície  $S$ , existe uma sequência finita de superfícies  $S_{i=1\dots N}$  tal que  $S_0 = \emptyset$  e a superfície  $S_i$  é obtida anexando um *handle*  $H_\lambda = (A_\lambda, B_\lambda)$  ao bordo de  $S_{i-1}$ .

# 4. Teoria de *Handlebody*

---

**Teorema.** (Decomposição de Superfícies por *Handlebodies*)  
Para toda superfície  $S$ , existe uma sequência finita de superfícies  $S_{i=1\dots N}$  tal que  $S_0 = \emptyset$  e a superfície  $S_i$  é obtida anexando um *handle*  $H_\lambda = (A_\lambda, B_\lambda)$  ao bordo de  $S_{i-1}$ .

$$S_0 = \emptyset$$

# 4. Teoria de *Handlebody*

---

**Teorema.** (Decomposição de Superfícies por *Handlebodies*)  
Para toda superfície  $S$ , existe uma sequência finita de superfícies  $S_{i=1\dots N}$  tal que  $S_0 = \emptyset$  e a superfície  $S_i$  é obtida anexando um *handle*  $H_\lambda = (A_\lambda, B_\lambda)$  ao bordo de  $S_{i-1}$ .

$$S_0 = \emptyset$$

$$S_1 = S_0 + \mathbf{H}_0$$





# 4. Teoria de *Handlebody*

**Teorema.** (Decomposição de Superfícies por *Handlebodies*)  
Para toda superfície  $S$ , existe uma sequência finita de superfícies  $S_{i=1\dots N}$  tal que  $S_0 = \emptyset$  e a superfície  $S_i$  é obtida anexando um *handle*  $H_\lambda = (A_\lambda, B_\lambda)$  ao bordo de  $S_{i-1}$ .

$$S_0 = \emptyset$$

$$S_1 = S_0 + \mathbf{H}_0$$

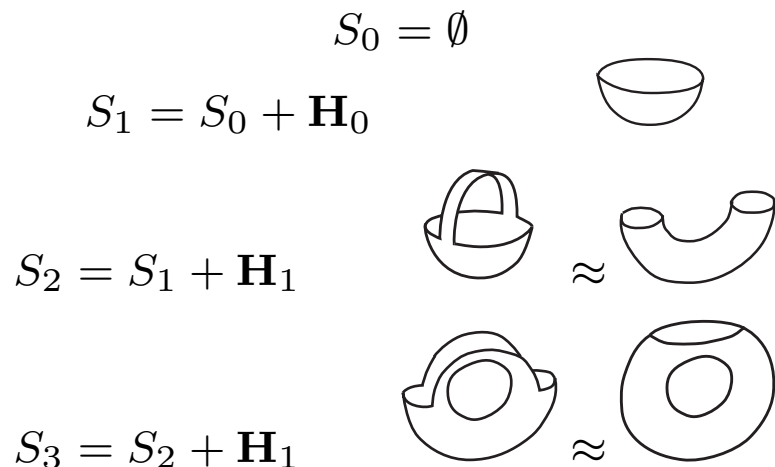


$$S_2 = S_1 + \mathbf{H}_1$$



# 4. Teoria de *Handlebody*

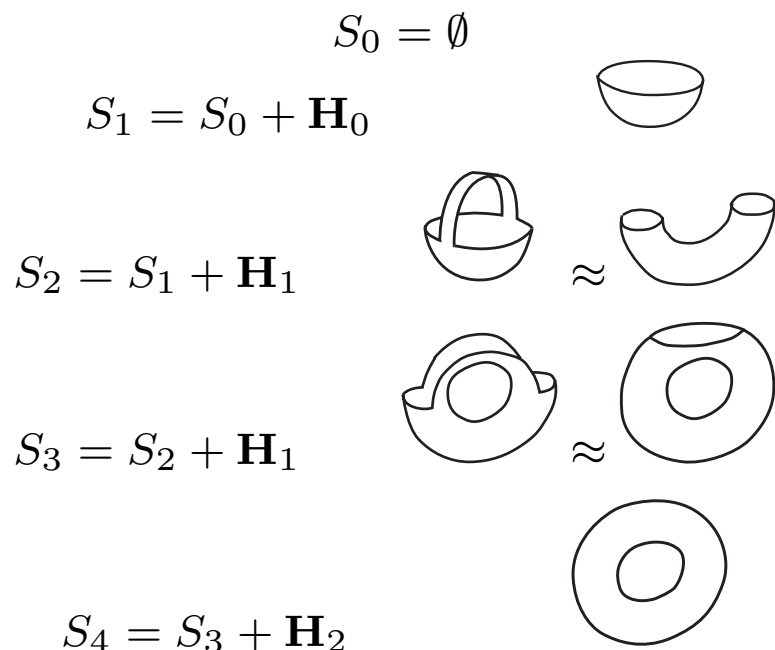
**Teorema.** (Decomposição de Superfícies por *Handlebodies*)  
Para toda superfície  $S$ , existe uma sequência finita de superfícies  $S_{i=1\dots N}$  tal que  $S_0 = \emptyset$  e a superfície  $S_i$  é obtida anexando um *handle*  $H_\lambda = (A_\lambda, B_\lambda)$  ao bordo de  $S_{i-1}$ .



# 4. Teoria de *Handlebody*

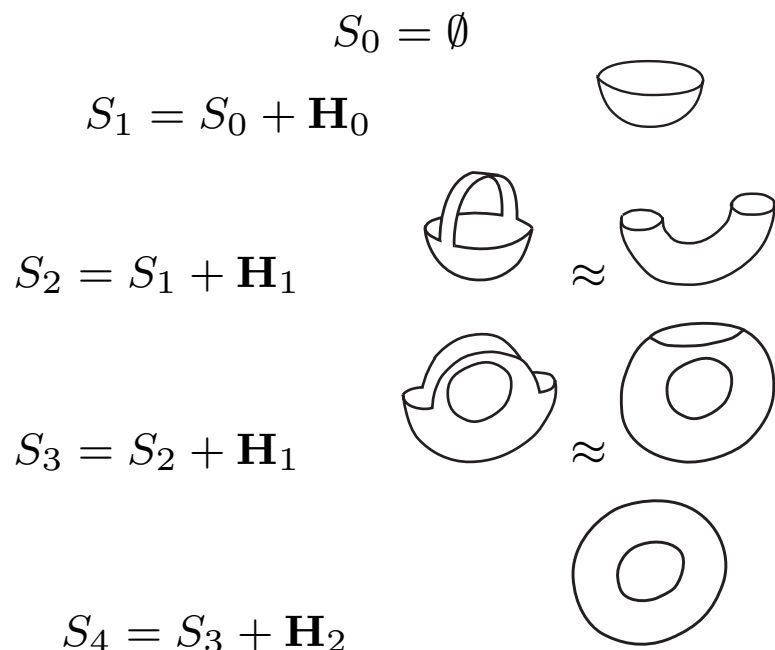
**Teorema.** (Decomposição de Superfícies por *Handlebodies*)

Para toda superfície  $S$ , existe uma sequência finita de superfícies  $S_{i=1\dots N}$  tal que  $S_0 = \emptyset$  e a superfície  $S_i$  é obtida anexando um *handle*  $H_\lambda = (A_\lambda, B_\lambda)$  ao bordo de  $S_{i-1}$ .



# 4. Teoria de *Handlebody*

**Teorema.** (Decomposição de Superfícies por *Handlebodies*)  
Para toda superfície  $S$ , existe uma sequência finita de superfícies  $S_{i=1\dots N}$  tal que  $S_0 = \emptyset$  e a superfície  $S_i$  é obtida anexando um *handle*  $H_\lambda = (A_\lambda, B_\lambda)$  ao bordo de  $S_{i-1}$ .



**Como detectar e implementar essas mudanças topológicas?**

# 4. Teoria de *HandleBody*

---

- A malha é representada por  $M = (V, E, F, B)$ ;

# 4. Teoria de *HandleBody*

---

- A malha é representada por  $M = (V, E, F, B)$ ;
- Com informações de vizinhança baseada na estrutura *half winged edge*;

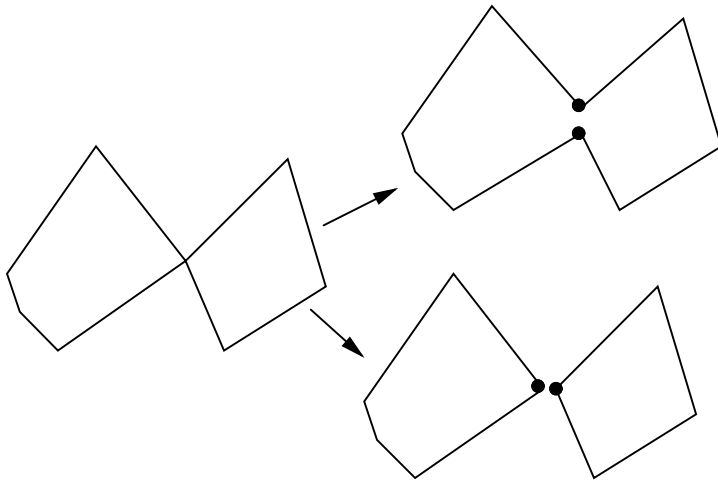
# 4. Teoria de *HandleBody*

---

- A malha é representada por  $M = (V, E, F, B)$ ;
- Com informações de vizinhança baseada na estrutura *half winged edge*;
- Diferenciação entre *pontos* e *vértices* para evitar ambiguidades;

# 4. Teoria de *HandleBody*

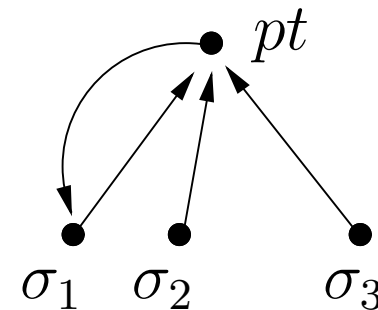
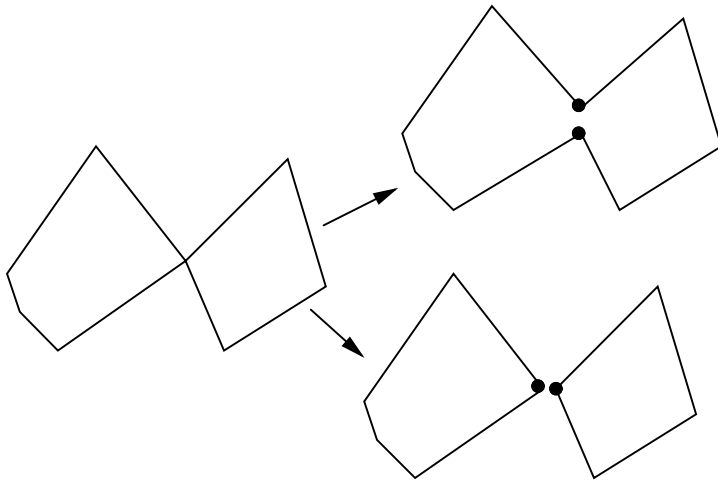
- A malha é representada por  $M = (V, E, F, B)$ ;
- Com informações de vizinhança baseada na estrutura *half winged edge*;
- Diferenciação entre *pontos* e *vértices* para evitar ambiguidades;





# 4. Teoria de *HandleBody*

- A malha é representada por  $M = (V, E, F, B)$ ;
- Com informações de vizinhança baseada na estrutura *half winged edge*;
- Diferenciação entre *pontos* e *vértices* para evitar ambiguidades;



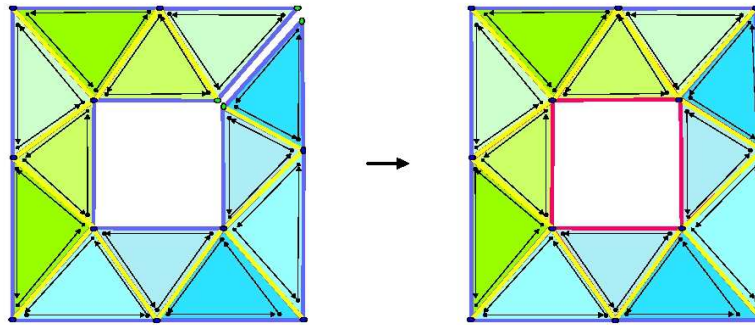
# 5. Operadores de *Handle*

---

**Definição.** Duas arestas  $e_{ij}$  e  $e_{kl}$  são topologicamente não-coincidentes se  $\#(\{\sigma_i, \sigma_j\} \cap \{\sigma_k, \sigma_l\}) = 0$

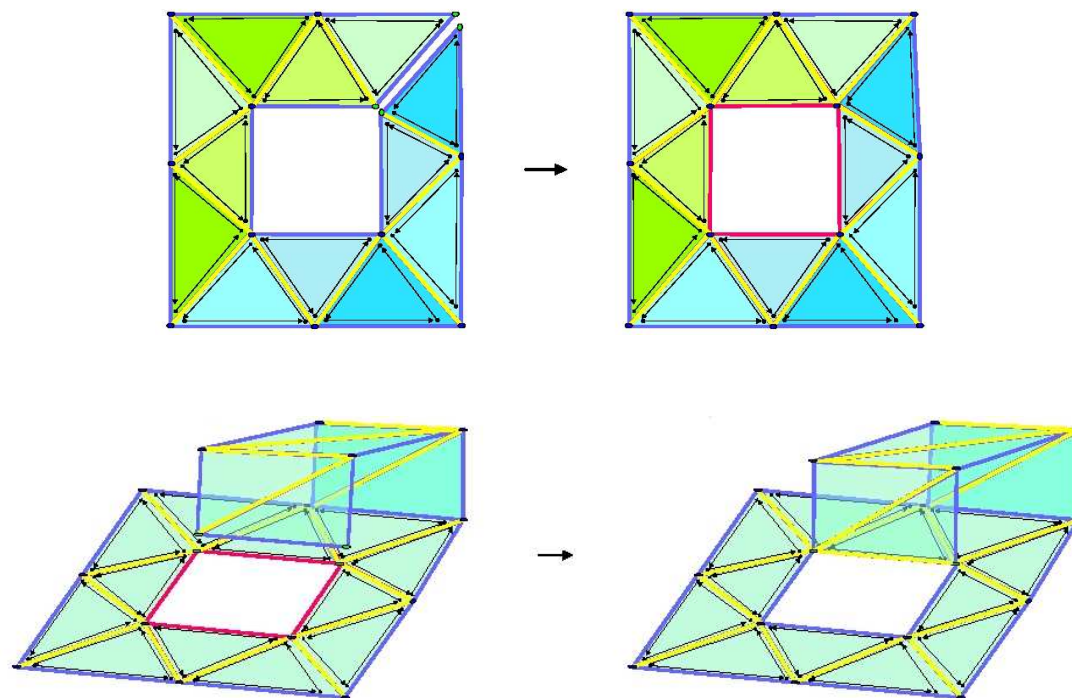
# 5. Operadores de *Handle*

**Definição.** Duas arestas  $e_{ij}$  e  $e_{kl}$  são topologicamente não-coincidentes se  $\#(\{\sigma_i, \sigma_j\} \cap \{\sigma_k, \sigma_l\}) = 0$



# 5. Operadores de *Handle*

**Definição.** Duas arestas  $e_{ij}$  e  $e_{kl}$  são topologicamente não-coincidentes se  $\#(\{\sigma_i, \sigma_j\} \cap \{\sigma_k, \sigma_l\}) = 0$





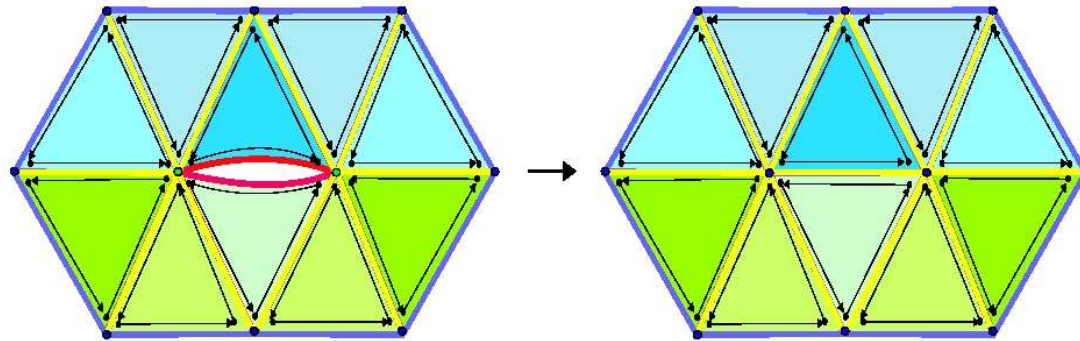
# 5. Operadores de *Handle*

---

**Definição.** Duas arestas  $e_{ij}$  e  $e_{kl}$  topologicamente coincidentes se elas têm a mesma topologia, isto é,  $\{\sigma_i, \sigma_j\} = \{\sigma_k, \sigma_l\}$ .

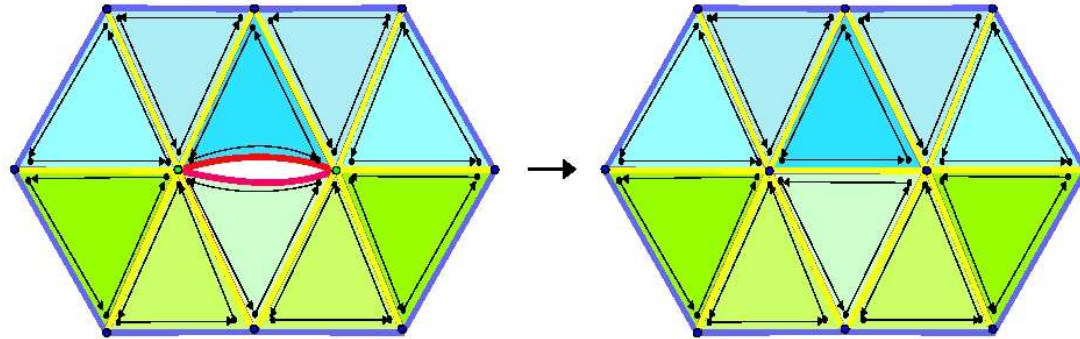
# 5. Operadores de *Handle*

**Definição.** Duas arestas  $e_{ij}$  e  $e_{kl}$  topologicamente coincidentes se elas têm a mesma topologia, isto é,  $\{\sigma_i, \sigma_j\} = \{\sigma_k, \sigma_l\}$ .



# 5. Operadores de *Handle*

**Definição.** Duas arestas  $e_{ij}$  e  $e_{kl}$  topologicamente coincidentes se elas têm a mesma topologia, isto é,  $\{\sigma_i, \sigma_j\} = \{\sigma_k, \sigma_l\}$ .



**Op. de *Handle* do tipo-2: identifica duas arestas topologicamente coincidentes.**



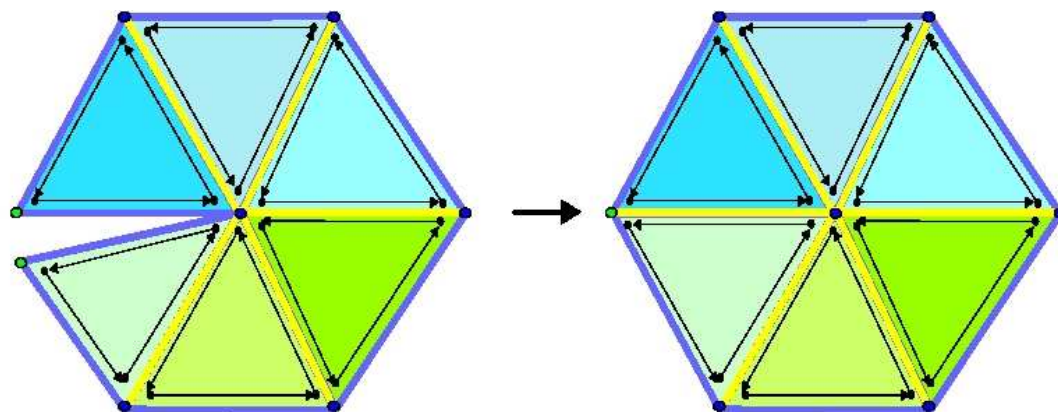
# 5. Operadores de *Handle*

---

**Definição.** Duas arestas  $e_{ij}$  e  $e_{kl}$  são topologicamente semi-coincidentes se  $\#(\{\sigma_i, \sigma_j\} \cap \{\sigma_k, \sigma_l\}) = 1$ .

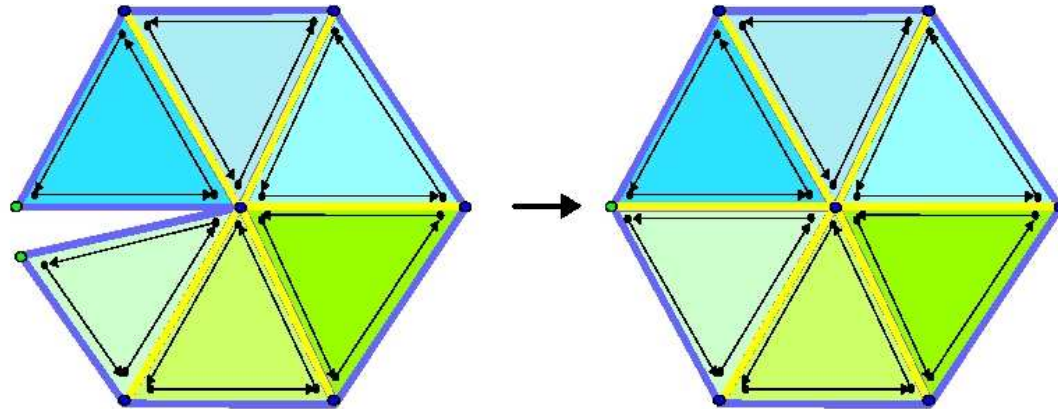
# 5. Operadores de *Handle*

**Definição.** Duas arestas  $e_{ij}$  e  $e_{kl}$  são topologicamente semi-coincidentes se  $\#(\{\sigma_i, \sigma_j\} \cap \{\sigma_k, \sigma_l\}) = 1$ .



# 5. Operadores de *Handle*

**Definição.** Duas arestas  $e_{ij}$  e  $e_{kl}$  são topologicamente semi-coincidentes se  $\#(\{\sigma_i, \sigma_j\} \cap \{\sigma_k, \sigma_l\}) = 1$ .



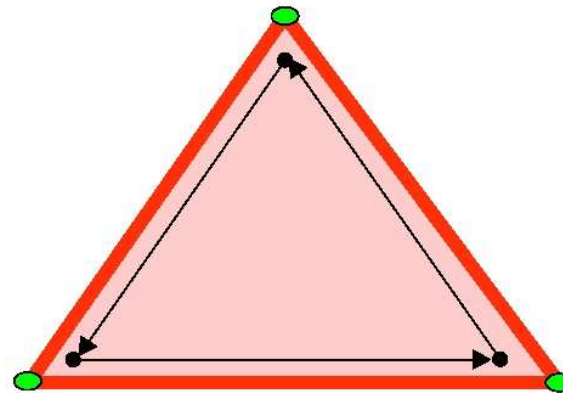
**Homeomorfismos: identifica duas arestas topologicamente semi-coincidentes.**

# 5. Operadores de *Handle*

---

Operador de *handle* tipo-0:

NIL



# 5. Operadores de *Handle*

---

API's:

- **criar**( $p_i$ ,  $p_j$ ,  $p_k$ )  
*handle* tipo-0;

# 5. Operadores de *Handle*

---

API's:

- **criar**( $p_i, p_j, p_k$ )  
*handle* tipo-0;
- **colar**( $e_{ij}, e_{ji}$ );  
*handle* tipo-1;  
*handle* tipo-2;  
homeomorfismos;

# 5. Operadores de *Handle*

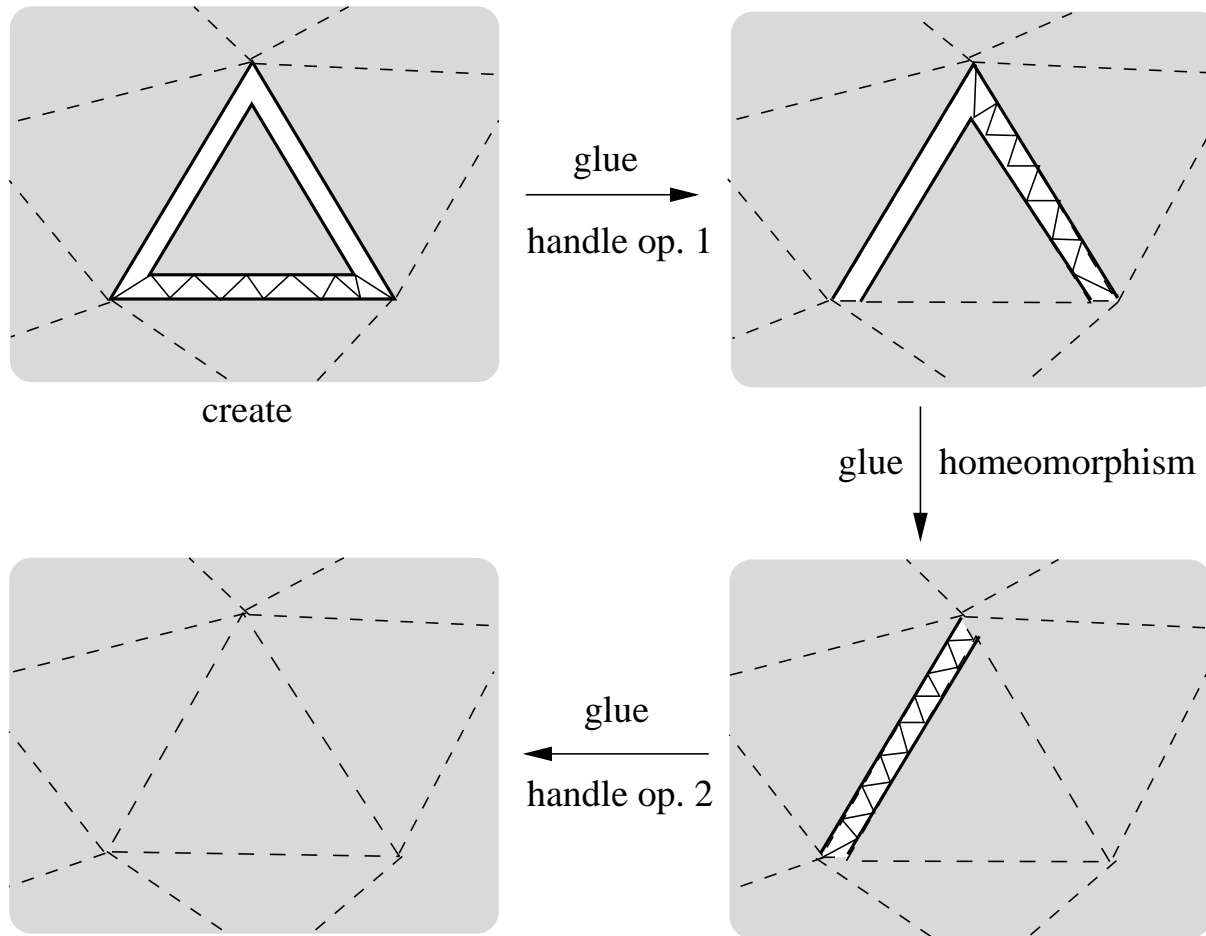
---

API's:

- **criar**( $p_i, p_j, p_k$ )  
*handle* tipo-0;
- **colar**( $e_{ij}, e_{ji}$ );  
*handle* tipo-1;  
*handle* tipo-2;  
homeomorfismos;
- Elas irão imitar o processo de construção de superfícies do teorema;

# 5. Operadores de *Handle*

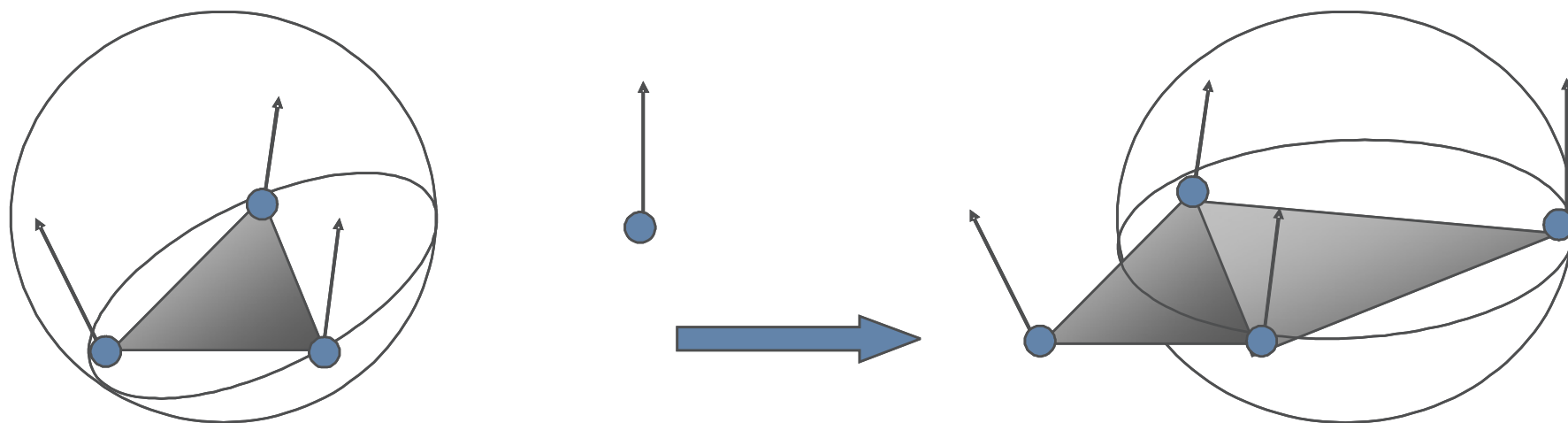
Sequência de Operações:





# 6. O Algoritmo

Critério geométrico:



# 6. O Algoritmo

---

Operações topológicas

# 6. O Algoritmo

---

Operações topológicas

+

Operações Geométricas (Ball-Pivoting)

# 6. O Algoritmo

---

Operações topológicas

+

Operações Geométricas (Ball-Pivoting)

=

Algoritmo Ball-Pivoting

# 6. O Algoritmo

---

Operações topológicas:

- criar;
- colar;

# 6. O Algoritmo

---

Operações topológicas:

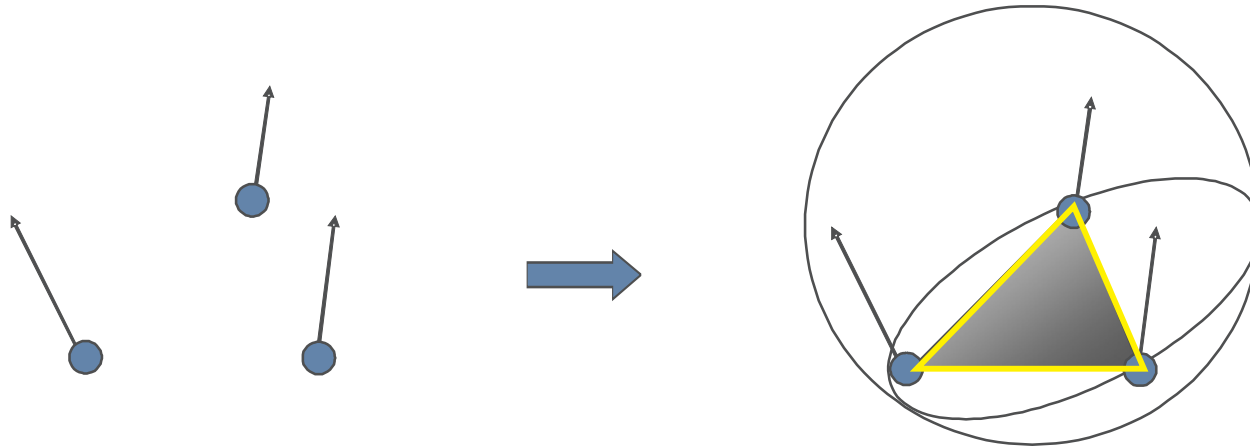
- criar;
- colar;

Operações geométricas:

- primeira-escolha;
- passo-principal;

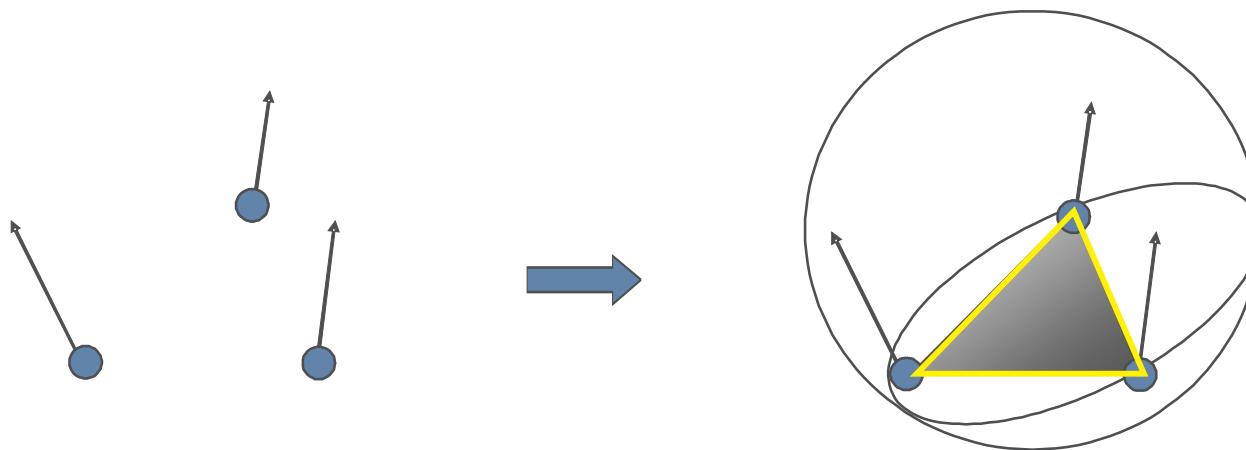
# 6. Algoritmo

Primeira-Escolha:



# 6. Algoritmo

Primeira-Escolha:

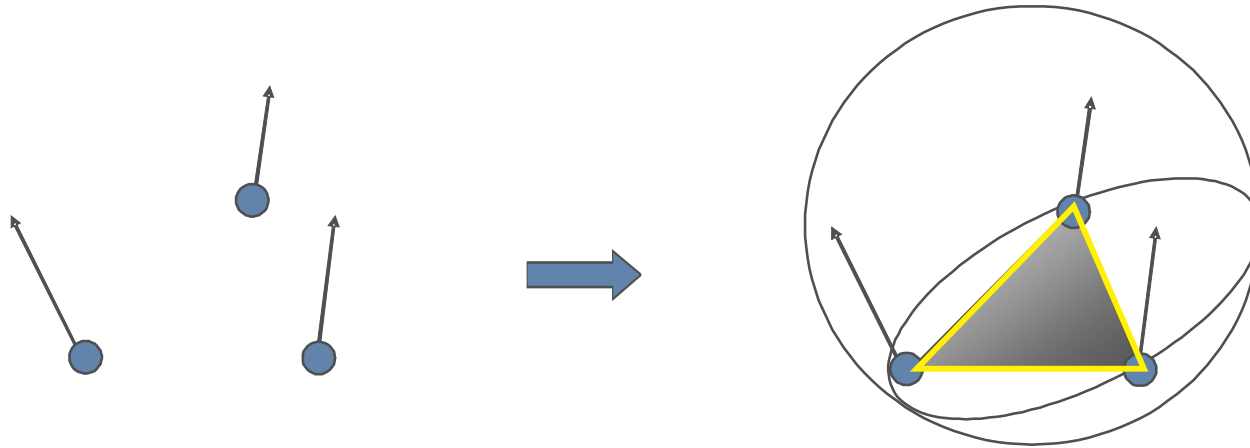


- Mil e uma estratégias. Detalhes nas notas;



# 6. Algoritmo

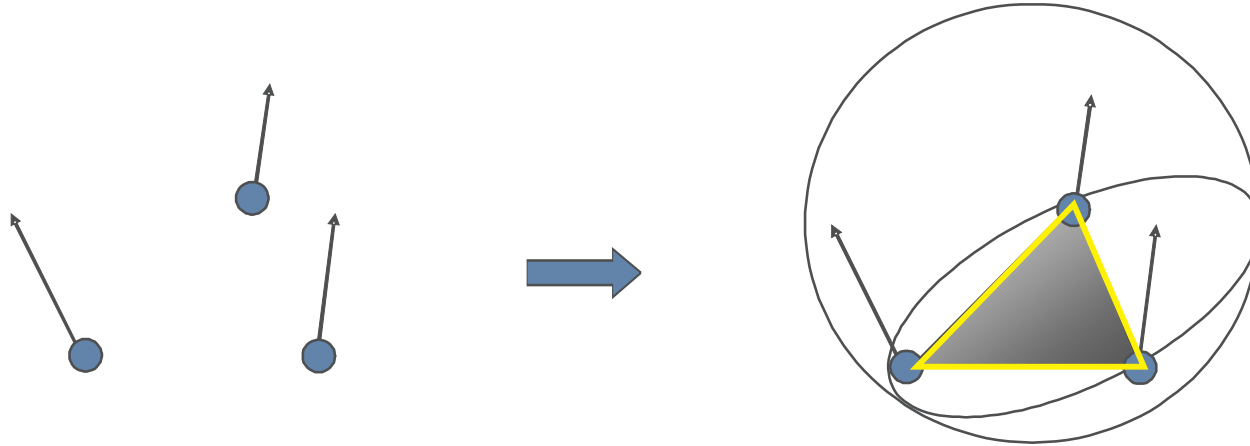
Primeira-Escolha:



- Mil e uma estratégias. Detalhes nas notas;
- Uma chamada a cada componente conexa;

# 6. Algoritmo

Primeira-Escolha:

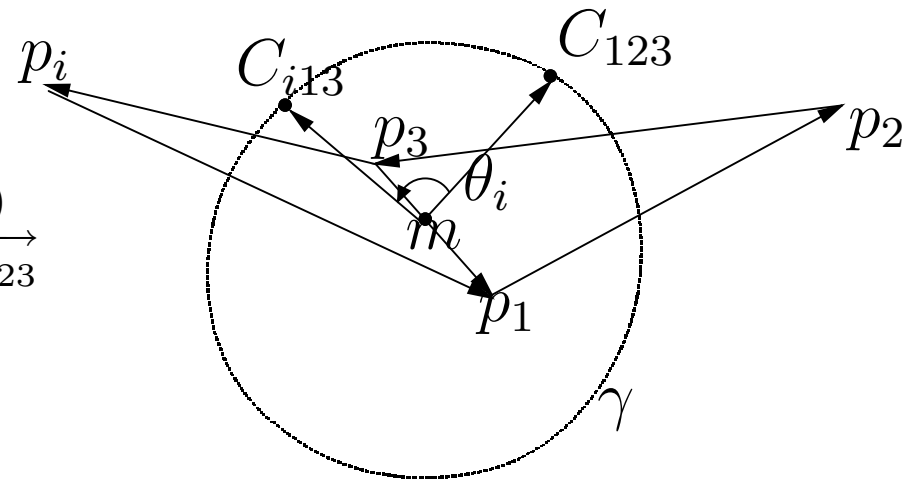


- Mil e uma estratégias. Detalhes nas notas;
- Uma chamada a cada componente conexa;
- Calcular o centro da bola  $B_\alpha(\Delta)$ .

# 6. O Algoritmo

## Passo-Principal:

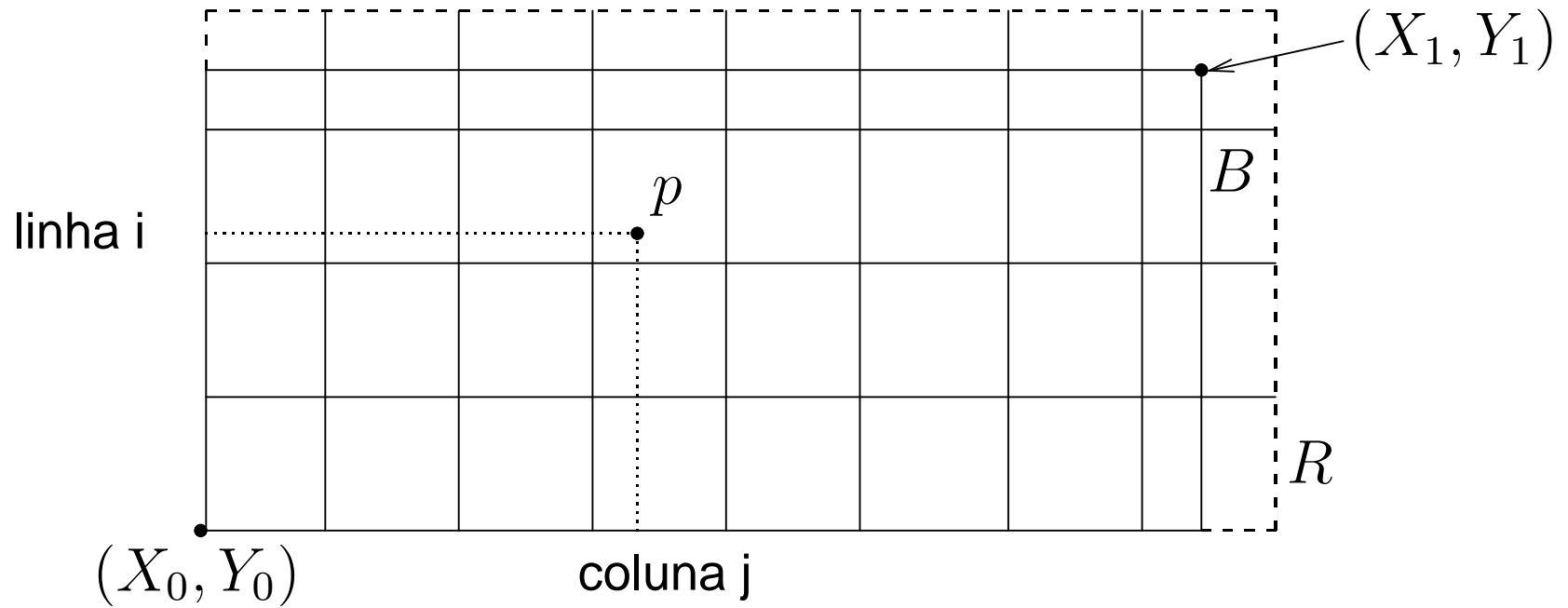
- 1: Entrada( $M, B_\alpha(p_1p_2p_3), m, C_{123}$ )
- 2: Saída ( $p_0 \in V(m)$  tal que  $C_{123}\hat{m}C_{031}$  é mínimo)
- 3:  $p_0 \leftarrow NULO$
- 4:  $minang \leftarrow 2\pi$
- 5: **for all**  $p_i \in V(m)$  **do**
- 6:   calcule o centro de  $C_{i13}$  de  $B_\alpha(p_i p_1 p_3)$
- 7:   calcule o angulo  $\theta_i$  entre  $\overrightarrow{mC_{123}}$  e  $\overrightarrow{mC_{i23}}$
- 8:   **if**  $\theta_i < minang$  **then**
- 9:      $p_0 \leftarrow p_i$
- 10:   **end if**
- 11: **end for**
- 12: return  $p_0$



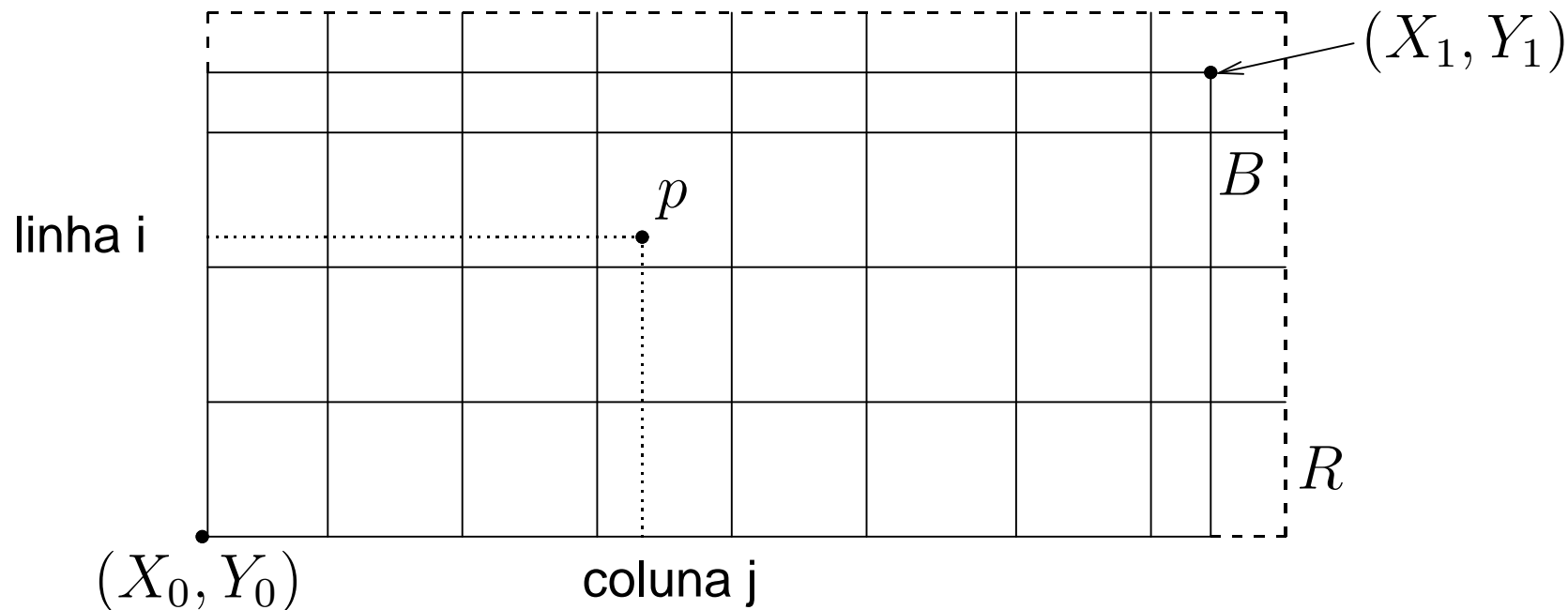
# 6. O Algoritmo

```
1: while (verdadeiro) do
2:   while (( $e_{ij} \leftarrow$  primeiro elemento de  $Q \neq NULO$ ) do
3:      $p_k \leftarrow$  passo-principal( $e_{ij}$ )
4:     if ( $p_k \neq \emptyset$ ) e ( $nao\_usado(p_k)$  ou  $indice(p_k)$ ) then
5:        $criar(p_i, p_j, p_k)$ 
6:        $colar(e_{ij}, e_{ji})$ 
7:       if  $e_{ki} \in B$  then
8:          $colar(e_{ik}, e_{ki})$ 
9:       end if
10:      if  $e_{jk} \in B$  then
11:         $colar(e_{kj}, e_{jk})$ 
12:      end if
13:      else
14:         $marque-como-bordo(e_{ij})$ 
15:      end if
16:   end while
17:   if ( $p_i, p_j, p_k$ )= $primeira-escolha()$  then
18:      $criar(p_i, p_j, p_k)$ 
19:   else
20:     break;
21:   end if
22: end while
```

# 7. Aceleração

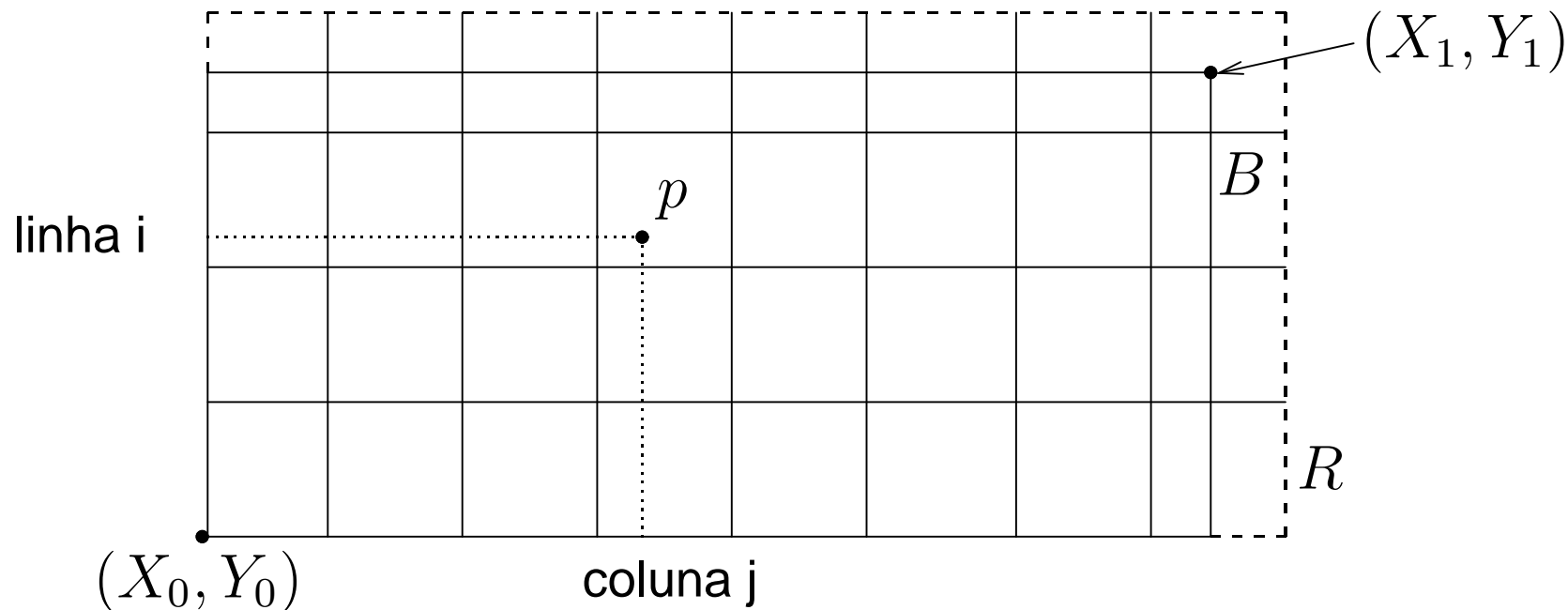


# 7. Aceleração



- Subdivisão uniforme do espaço em voxels de lado  $2\alpha$ ;

# 7. Aceleração



- Subdivisão uniforme do espaço em voxels de lado  $2\alpha$ ;
- No caso tridimensional, teríamos uma estrutura de matriz  $M$  de voxels.

# 8. Conclusão

---

- Revisamos o ABP dentro de um *framework* topológico;



# 8. Conclusão

---

- Revisamos o ABP dentro de um *framework* topológico;
- É razoavelmente simples de implementar;

# 8. Conclusão

---

- Revisamos o ABP dentro de um *framework* topológico;
- É razoavelmente simples de implementar;
- ...

---

Demo.