Rasterização de Primitivas

Linhas e Polígonos

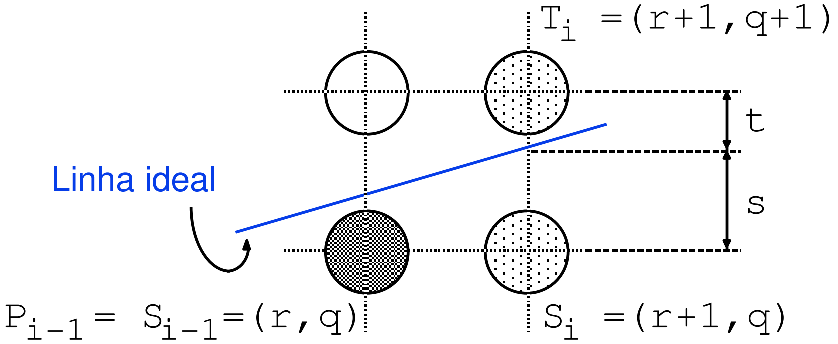
Fernando Birra

Rasterização de Segmentos de Reta

ALGORITMO DE BRESENHAM

Método eficiente [1965] **por uso exclusivo de aritmética de inteiros.**

Como escolher o pixel no passo i **?**



P_i = **if** $s - t < 0$ **then** S_i
else T_i

Hipóteses:

$y = mx + b$, com $0 \leq m \leq 1$

OBS.: O algoritmo é de grande importância histórica, mas a explicação que se segue é de carácter informativo e fora do programa de CGI.

M.Próspero

Rasterização de Segmentos de Reta

ALGORITMO DE BRESENHAM

Escolher a designação das coordenadas tal que: $x_1^2 + y_1^2 < x_2^2 + y_2^2$

Mudança de coordenadas:

$$\overline{(x_1, y_1)\ \ (x_2, y_2)} \rightarrow \overline{(0, 0)\ \ (dx, dy)} \quad \text{em que} \quad \begin{cases} dx = x_2 - x_1 \\ dy = y_2 - y_1 \end{cases}$$

Dedução das fórmulas finais:

$$y = x\ dy/dx \quad \overset{P_{i-1}=(r,q)}{\Rightarrow} \quad \begin{array}{l} s + q = (r+1)\ dy/dx \\ t = 1 - s = 1 + q - (r+1)\ dy/dx \end{array}$$

$$\Downarrow$$

$$s - t = 2\ (r+1)\ dy/dx - 2q - 1$$

Variável de decisão

$$d_i = dx\ (s - t) = 2\ dy\ (r+1) - 2\ q\ dx - dx$$

e como $\begin{cases} r = x_{i-1} \\ q = y_{i-1} \end{cases}$ virá:

$$d_{i+1} = 2\ dy\ (x_i + 1) - 2\ y_i\ dx - dx$$

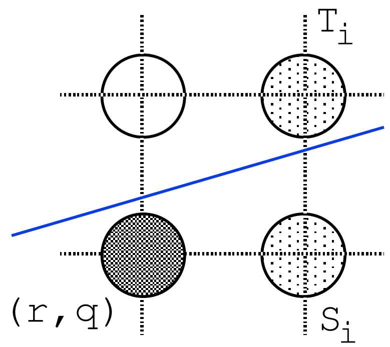
$$d_{i+1} - d_i = 2\ dy\ (x_i - x_{i-1}) - 2\ (y_i - y_{i-1})$$

$$\boxed{d_{i+1} = d_i + 2\ dy - 2\ (y_i - y_{i-1})\ dx}$$

M.Próspero

ALGORITMO DE BRESENHAM

$$d_{i+1} = d_i + 2\ dy - 2\ (y_i - y_{i-1})\ dx$$



P_i = if $d_i < 0$ then S_i

$y_i = y_{i-1} \quad \wedge \quad d_{i+1} = d_i + 2\ dy - 0 = d_i + k_s$

else T_i

$y_i = y_{i-1} + 1 \quad \wedge \quad d_{i+1} = d_i + 2\ dy - 2\ dx = d_i + k_t$

De $d_{i+1} = 2\ dy\ (x_i + 1) - 2\ y_i\ dx - dx$ resulta o valor inicial ($r=q=0$):

$$d_{i=1} = 2\ dy\ (0+1) - 0 - dx$$
$$= 2\ dy - dx$$

M.Próspero

ALGORITMO DE BRESENHAM

```
void LINE_BRESENHAM (int x1, int y1, int x2, int y2, int value){

   /* Condição de aplicabilidade:  0 ≤ m ≤ 1
     Para resolver nos outros octantes faz-se uma conversão para este! */

      int dx, dy, ks, kt, d, x, y, x_end;

      dx = abs(x2-x1);    dy = abs(y2-y1);    d = 2*dy - dx;
      ks = 2*dy;  kt = 2*(dy - dx);
      if (x1 > x2) {
            x = x2;  y = y2;  x_end = x1;  }
      else {
            x = x1;  y = y1;  x_end = x2;  }
      WRITE_PIXEL(x, y, value);
      while (x < x_end) {
            x++;
            if (d < 0)
                  d += ks;
            else {
                  y++;  d += kt;
            }
            WRITE_PIXEL(x, y, value);
      }
}
```

M.Próspero

Rasterização de Segmentos de Reta

ALGORITMO DO PONTO MÉDIO **"MIDPOINT ALGORITHM"** [Pitteway 65 / Van Aken 85]

Justificação: **O algoritmo de Bresenham não é de generalização fácil para as cónicas em geral.**

Aplicação ao caso de uma reta, cuja equação é

$$F(x,y) = a\,x + b\,y + c = 0$$

Comparando com a forma explícita ($0 \le m \le 1$ **)**

$$y = m\,x + B = (dy/dx)\,x + B$$

determinam-se os coeficientes da forma implícita:

$$F(x,y) = dy\,x - dx\,y + B\,dx = 0 \quad \Longrightarrow \quad \boxed{\begin{aligned} a &= dy \\ b &= -\,dx \\ c &= B\,dx \end{aligned}}$$

Hipóteses: $0 \le m \le 1$ e $dy \ge 0$, por escolha da ordenação dos pontos que definem o segmento de reta.

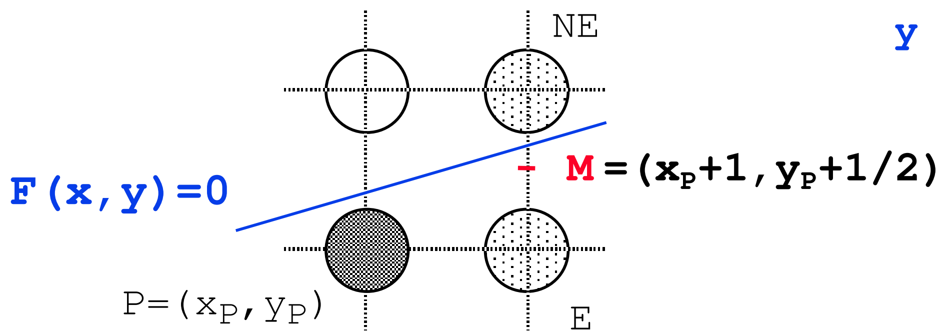
Exercício: Qual o sinal de $F(x,y)$ para um ponto acima da reta? E abaixo?

M.Próspero

Rasterização de Segmentos de Reta

ALGORITMO DO PONTO MÉDIO

$y = mx + b$, com $0 \le m \le 1$



Comparação no ponto médio M para se conhecer o novo ponto P com abcissa x_P+1 :

```
P  =  if  F(M) < 0  then  E
                    else  NE
```

$$F(M) = a(x_P + 1) + b(y_P + 1/2) + c = d'$$

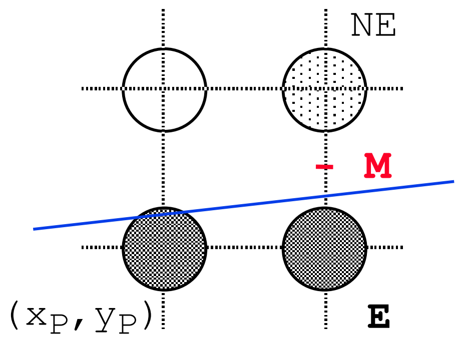
Variável de decisão

```
P  =  if  d' < 0  then  E
                  else  NE
```

M.Próspero

ALGORITMO DO PONTO MÉDIO

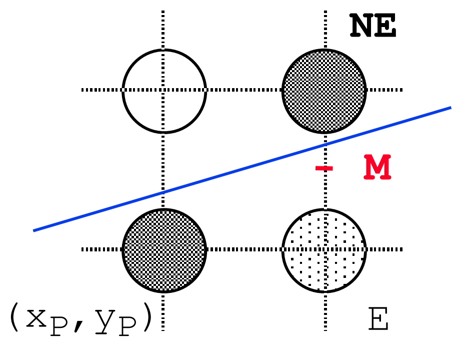
Como $d' = F(M)$**, quando…**



… a escolha for E (novo **M**, para x_P+2, manterá a ordenada)**, no passo seguinte é:**

$$d' = F(x_P+2, y_P+1/2) = a(x_P+2) + b(y_P+1/2) + c$$

ou seja, numa instrução de afetação: $\boxed{d' = d' + a}$



… a escolha for NE (a ordenada do novo **M**, para x_P+2, será incrementada)**, no passo seguinte é:**

$$d' = F(x_P+2, y_P+3/2) = a(x_P+2) + b(y_P+3/2) + c$$

ou seja, numa instrução de afetação: $\boxed{d' = d' + a + b}$

Como, por hipótese, as coordenadas das extremidades do segmento de reta são valores inteiros, **a** e **b** também o são.

Inicialização da variável de decisão no ponto $P(x_1, y_1)$ **:**

$$\begin{aligned} F(x_1+1, y_1+1/2) &= a(x_1+1) + b(y_1+1/2) + c \\ &= a\ x_1 + b\ y_1 + c + a + b/2 \\ &= F(x_1, y_1) + a + b/2 \\ &= a + b/2 \end{aligned}$$

M.Próspero

Rasterização de Segmentos de Reta

ALGORITMO DO PONTO MÉDIO

O valor inicial da variável de decisão seria:

!

$$d' = F(x_1+1, y_1+1/2) = a + b/2$$

$a = dy$
$b = -dx$

Mas, para se trabalhar exclusivamente com valores inteiros, multiplique-se por 2 e substitua-se 2d′ por d:

$$d = 2a + b = 2dy - dx$$

Conclusão:

```
P =  if  d < 0  then { E }
        x = x + 1   ∧   d = d + 2 * dy
                    else { NE }
        x = x + 1 ;  y = y + 1  ∧  d = d + 2 * (dy − dx)
WRITE_PIXEL(x,y,value)
```

$d' = d' + a$

$d' = d' + a + b$

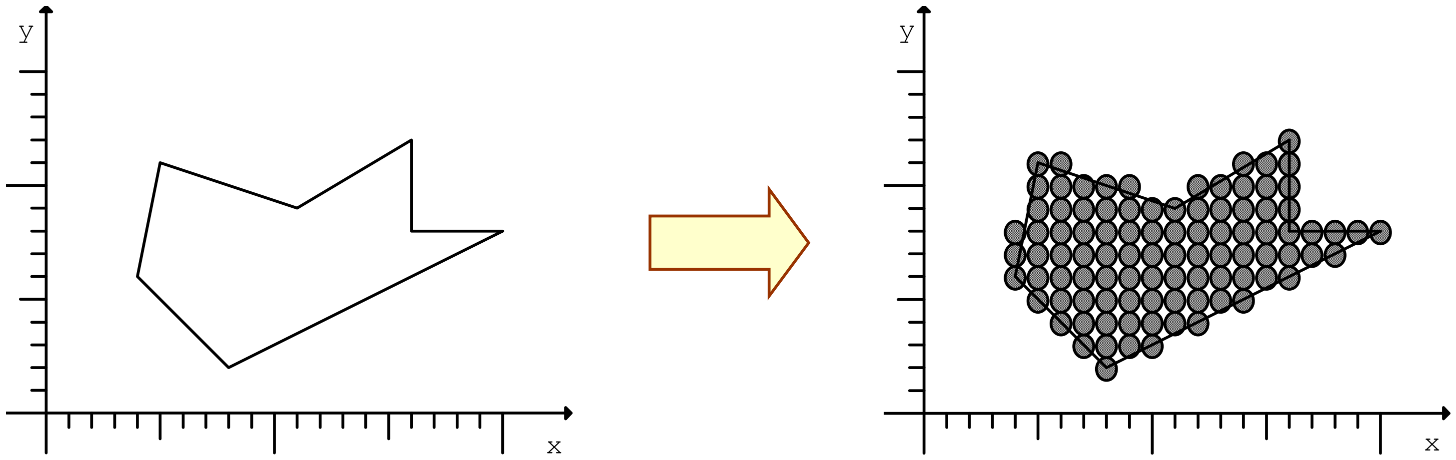
M.Próspero

POLÍGONOS

FILL AREA

Objetivo: **Preenchimento de um polígono definido pela <u>sequência</u> dos respetivos <u>vértices</u> (pontos 2D), independentemente dos valores pré-existentes na memória de refrescamento.**

Um exemplo concreto, numa 1.ª aproximação executada por preenchimento manual "a olho":



[mais adiante referido como **exemplo de referência** e que se verá não estar inteiramente correto!]

M.Próspero

FILL AREA

ALGORITMO PAR-ÍMPAR *(Even-Odd)*

É um algoritmo de VARRIMENTO por linhas (*scan-lines*), em que os pixels são testados, não individualmente, mas tirando partido da…

coerência por linha de varrimento

Se um dado pixel duma linha pertence ao polígono, é muito provável que os pixels vizinhos também pertençam.

coerência por aresta do polígono

Se uma aresta intersecta uma dada linha de varrimento, é muito provável que intersecte também as linhas vizinhas.

M.Próspero

Rasterização de Polígonos

Principais etapas do algoritmo de FILL AREA

Para cada linha de varrimento

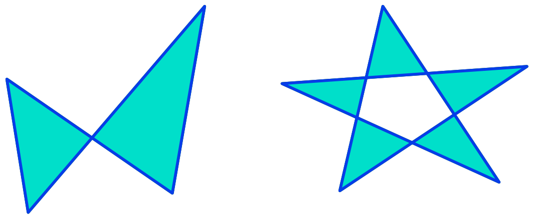
1 **calcular todas as intersecções com as arestas do polígono.**

As arestas horizontais são excluídas do algoritmo

2 **ordenar essas intersecções segundo os valores crescentes da abcissa.**

3 **preencher todos os pixels entre pares de intersecções.**

NB: **O algoritmo também funciona para polígonos com arestas que se cruzem (os pontos fronteira destes dois exemplos não foram gerados pelo algoritmo).**



M.Próspero

Rasterização de Polígonos

Is a given point P inside or outside a Polygon ?

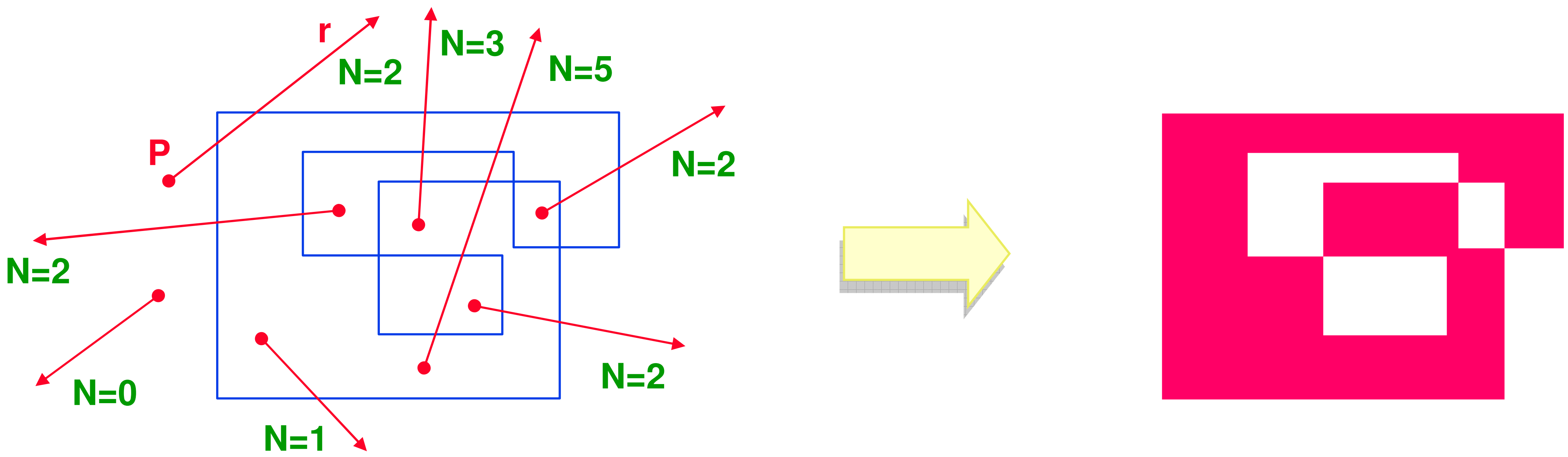
1 **Let r be a half of an arbitrary straight line starting at P.**

> Note: avoid passing r trough a vertex.

2 **Count the total number N of intersections with the edges of the polygon.**

3 **If N is <u>even</u> or 0 then P is <u>outside</u> the polygon**
else P is <u>inside</u> the polygon.

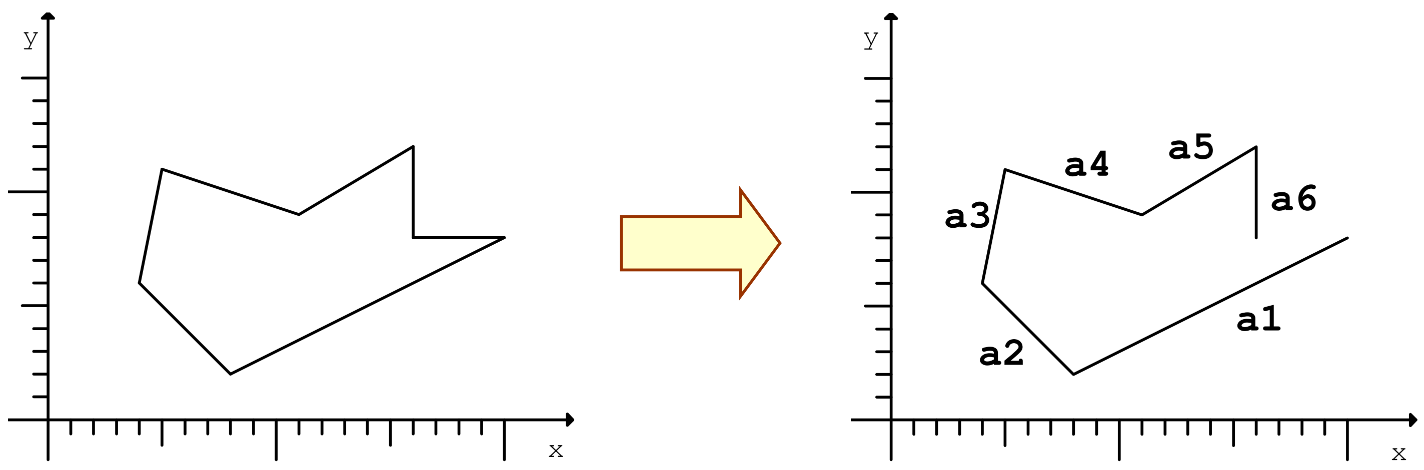
Examples:



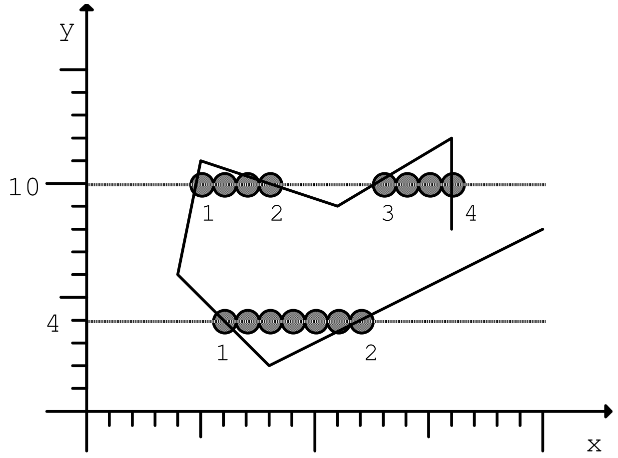
M.Próspero

Rasterização de Polígonos

Exemplo de exclusão de uma aresta, por ser horizontal:

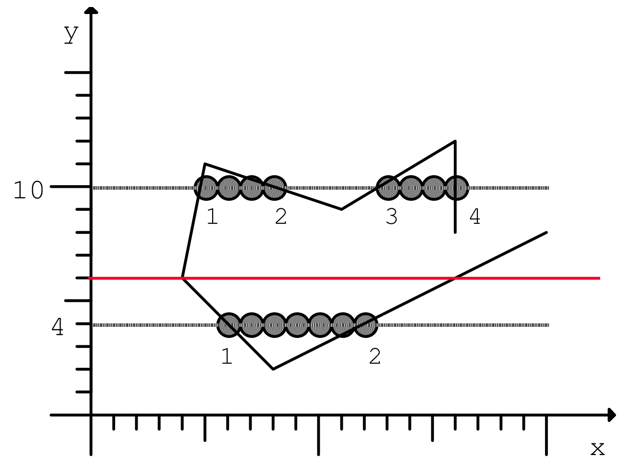


O preenchimento ao longo das linhas y=4 **e** y=10 **:**



M.Próspero

Rasterização de Polígonos



PROBLEMA NA PASSAGEM POR ALGUNS VÉRTICES

A linha y=6 dá origem a um número <u>ímpar</u> de intersecções,

pelo que o preenchimento não dará o resultado pretendido!

OBS: Nos mínimos e máximos locais não se coloca tal problema (p.ex. y=9 ou y=11).

RESOLUÇÃO

O preenchimento faz-se entre cada par de intersecções, mas <u>excluídas as arestas</u> para as quais a ordenada da linha de varrimento seja a sua <u>ordenada máxima</u>.

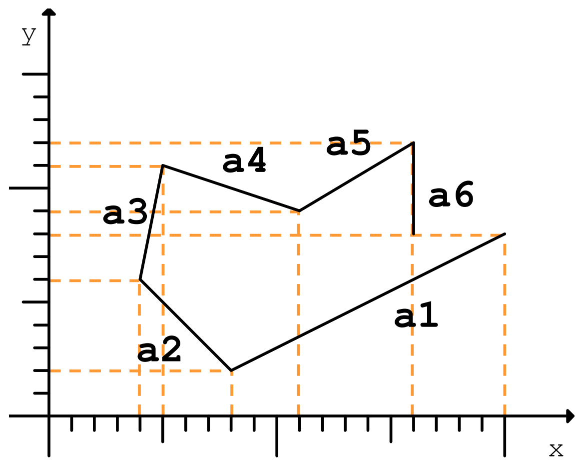
OBS: Para y=6, a aresta **a2** ficaria excluída, reduzindo-se as intersecções a duas (**a3** e **a1**).

M.Próspero

ESTRUTURAS DE DADOS

REGISTOS DAS ARESTAS

aresta	ymax	x	1/m	apontador
a1	8	8	2	
a2	6	8	-1	
a3	11	4	1/5	
a4	11	11	-3	
a5	12	11	5/3	
a6	12	16	0	



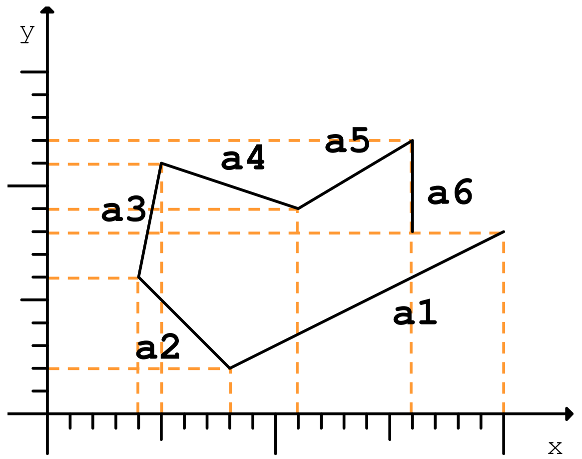
↑ Chave de identificação

↑ Abcissa da intersecção da aresta com a linha de varrimento (informação não estática, inicializada para a linha de ordenada mínima que intersecta essa aresta)

M.Próspero

Rasterização de Polígonos

TABELA DE ARESTAS (TA)



Linha y	Novas arestas intersectadas
0	–
1	–
2	**a1 → a2**
3	–
4	–
5	–
6	**a3**
7	–
8	**a6**
9	**a4 → a5**
10	–
11	–
12	–
...	–

Havendo mais do que um elemento em cada entrada, a respetiva lista de arestas ordena-se crescente-mente segundo as abcissas (Em cada um dos dois casos do exemplo acontece as abcissas terem valores iguais por partilha de vértice entre arestas, pelo que qualquer uma das duas listas poderia ter ordem contrária).

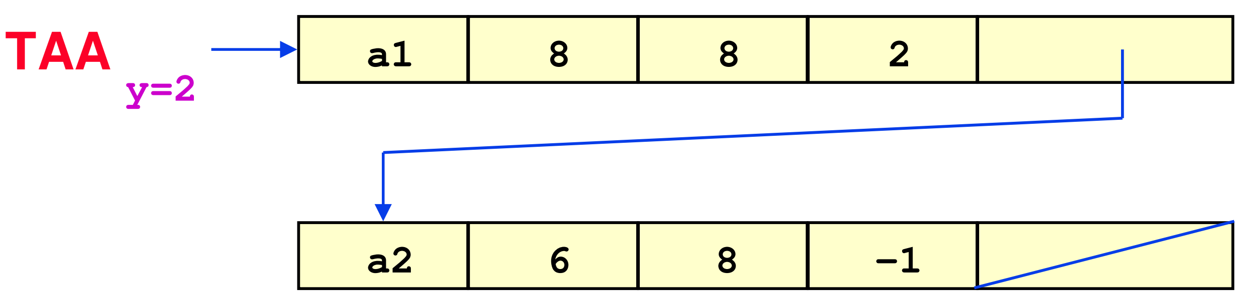
M.Próspero

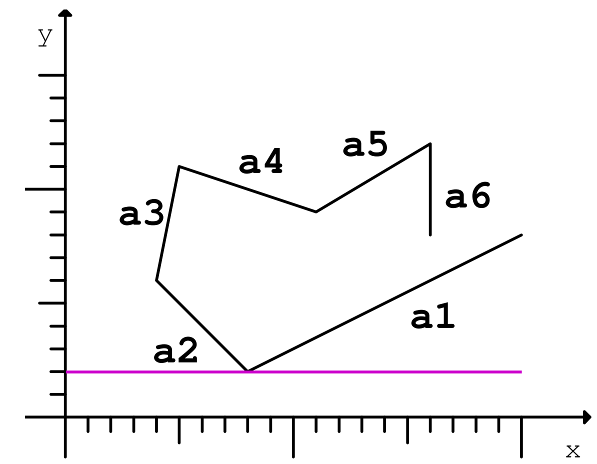
TABELA DE ARESTAS ATIVAS (TAA)

Estrutura dinâmica, sempre ordenada pelos valores de abcissa, para cada linha de varrimento.

Partindo de `y = mx + b`**, a atualização para a linha seguinte (coerência por arestas) pode obter-se com uma fórmula de recorrência:**

$$y_{i+1} - y_i = mx_{i+1} - mx_i = 1 \quad \rightarrow \quad x_{i+1} = x_i + 1/m$$





Introduzem-se na TAA as arestas lidas na TA para a linha de varrimento corrente.

M.Próspero

TABELA DE ARESTAS ATIVAS (TAA)

Estrutura dinâmica sempre ordenada pelos valores de abcissa, para cada linha de varrimento.

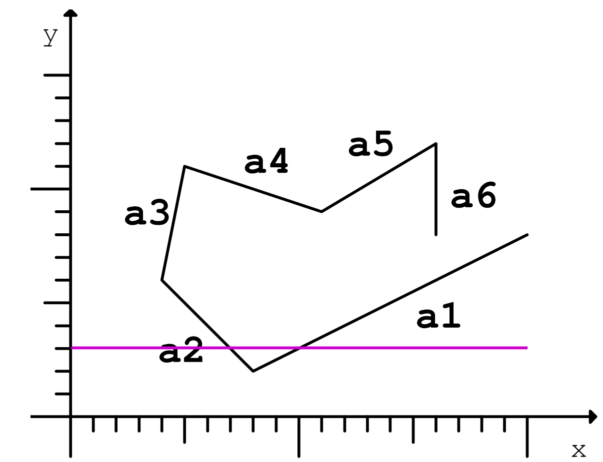
Passa-se para a linha seguinte (coerência por arestas) atualizando-se a abcissa

$$x_{i+1} = x_i + 1/m$$

TAA $_{y=3}$

a2	6	7	-1	

a1	8	10	2	



M.Próspero

Rasterização de Polígonos

TABELA DE ARESTAS ATIVAS (TAA)

Estrutura dinâmica sempre ordenada pelos valores de abcissa, para cada linha de varrimento.

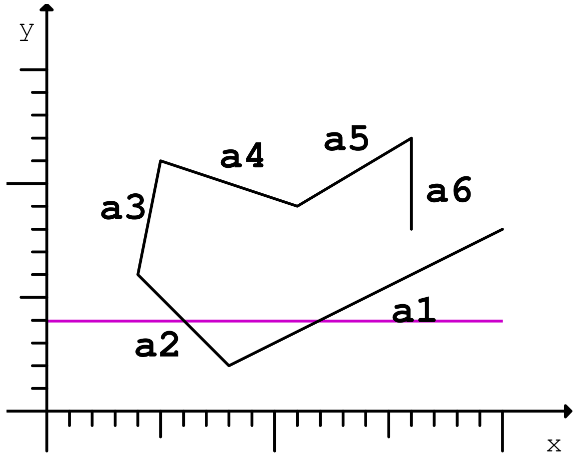
Passa-se para a linha seguinte (coerência por arestas) atualizando-se a abcissa

$$x_{i+1} = x_i + 1/m$$

$\text{TAA}_{y=4}$ →

a2	6	6	-1	→

a1	8	12	2	/



M.Próspero

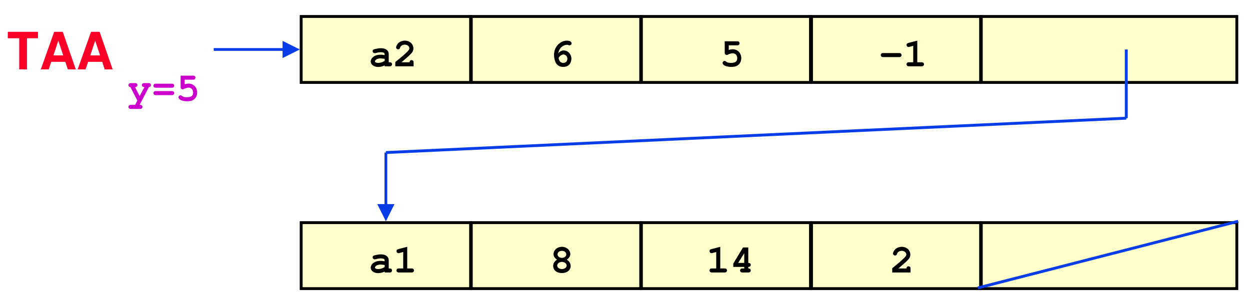
Rasterização de Polígonos

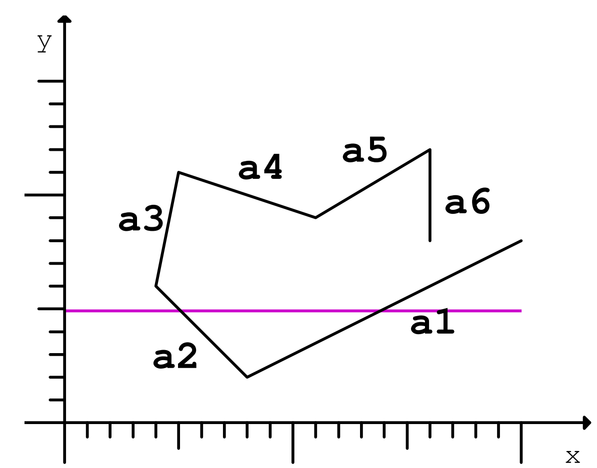
TABELA DE ARESTAS ATIVAS (TAA)

Estrutura dinâmica sempre ordenada pelos valores de abcissa, para cada linha de varrimento.

Passa-se para a linha seguinte (coerência por arestas) atualizando-se a abcissa

$$x_{i+1} = x_i + 1/m$$





M.Próspero

Rasterização de Polígonos

TABELA DE ARESTAS ATIVAS (TAA)

Estrutura dinâmica sempre ordenada pelos valores de abcissa, para cada linha de varrimento.

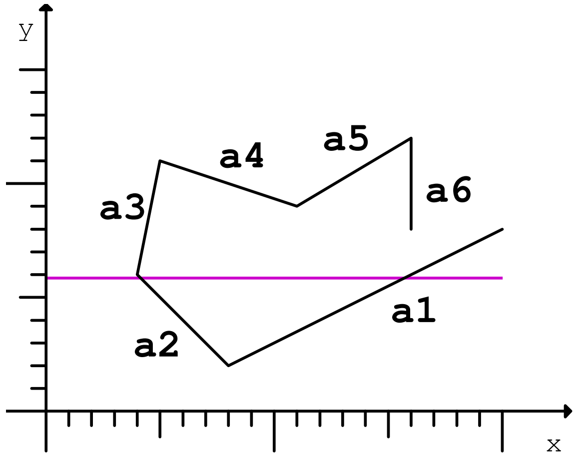
Passa-se para a linha seguinte (coerência por arestas) atualizando-se a abcissa

$$x_{i+1} = x_i + 1/m$$

$TAA_{y=6}$

a3	11	4	1/5	

a1	8	16	2	



M.Próspero

Rasterização de Polígonos

TABELA DE ARESTAS ATIVAS (TAA)

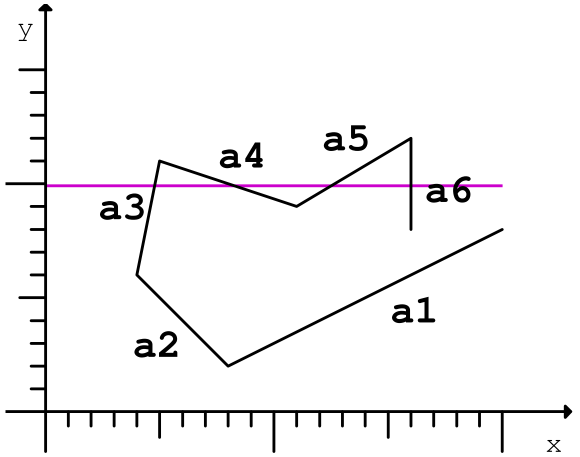
Estrutura dinâmica sempre ordenada pelos valores de abcissa, para cada linha de varrimento.

Passa-se para a linha seguinte (coerência por arestas) atualizando-se a abcissa

$$x_{i+1} = x_i + 1/m$$

$TAA_{y=10}$

a3	11	24/5	1/5	→
a4	11	8	-3	→
a5	12	38/3	5/3	→
a6	12	16	0	/



M.Próspero

Algoritmo de FILL AREA (*Even-Odd*)

- Construir TA (Seja Max a última entrada não vazia)
- Inicializar TAA com null
- Inicializar y com a primeira entrada na TA não vazia, caso exista, senão com Max+1

<u>Enquanto</u> TAA ≠ null <u>ou</u> y < Max+1

  - Eliminar de TAA as arestas com ymax==y
  - Atualizar TAA com as arestas referidas em TA[y] e ordenar a lista em x
  - Percorrer TAA para preencher as carreiras de pixels entre os valores de x de cada par de arestas
  - Incrementar y
  - Para cada aresta em TAA, fazer x=x+1/m

OBS.:
O algoritmo de FILL AREA atua sobre os pixels que se considera serem o *"interior"* do polígono. Pretendendo-se visualizar também as arestas, elas deverão ser rasterizadas como segmentos de reta imediatamente a seguir à aplicação de FILL AREA.

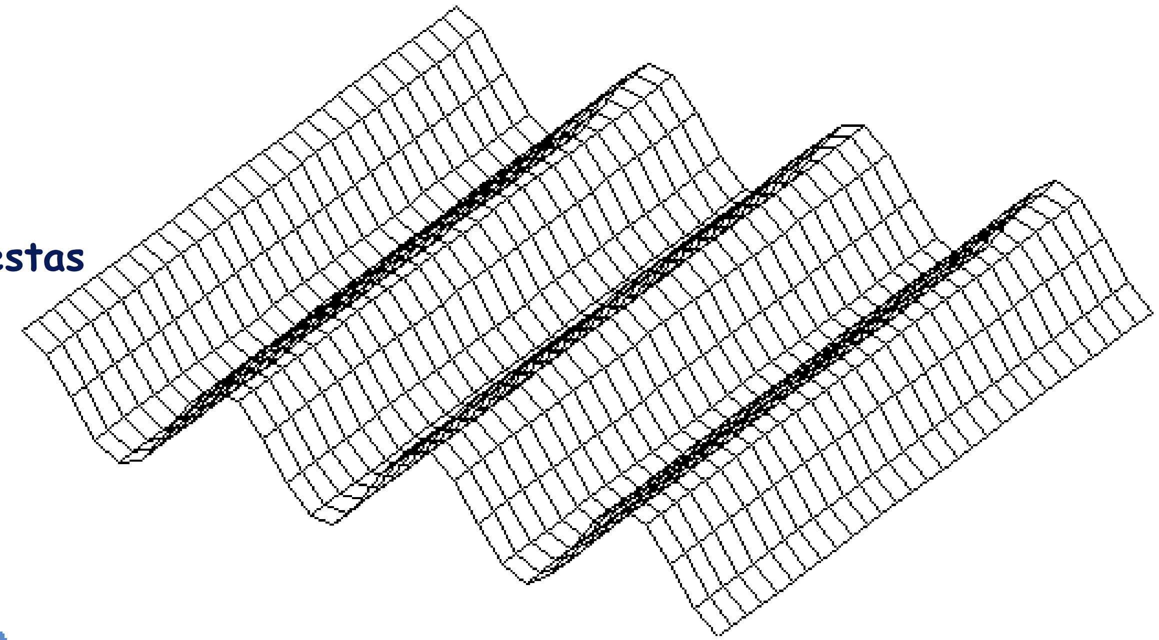
M.Próspero

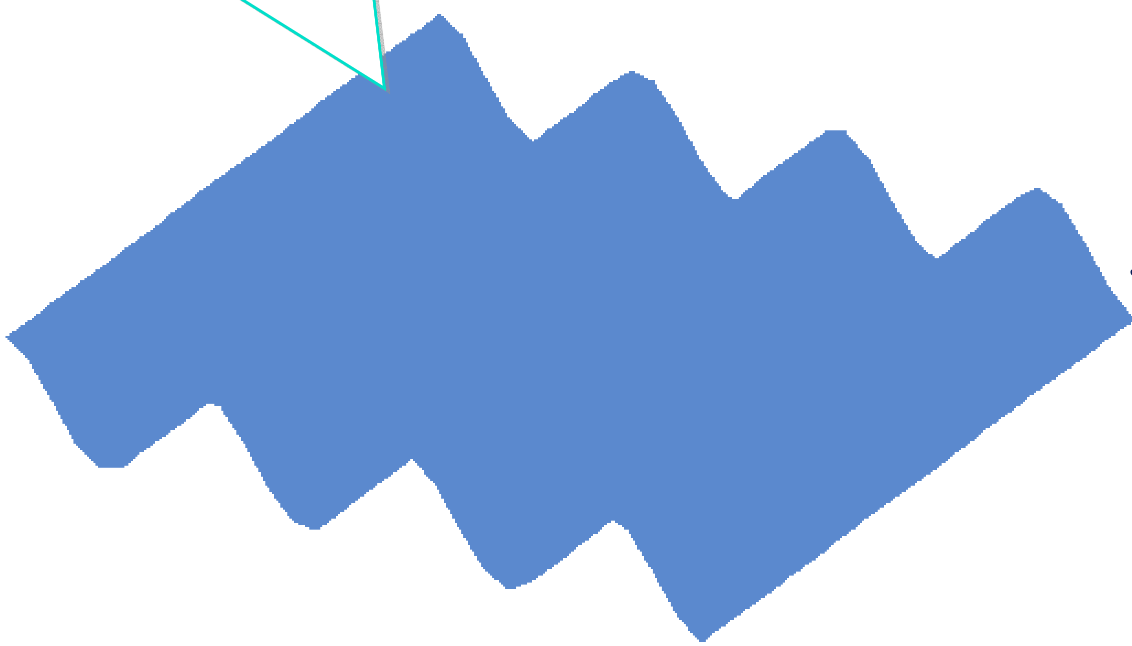
Rasterização de Polígonos

Aplicação em Malhas de Polígonos

Para se obter este resultado, o algoritmo não só deverá pintar o interior de cada polígono, como também parte da sua fronteira!

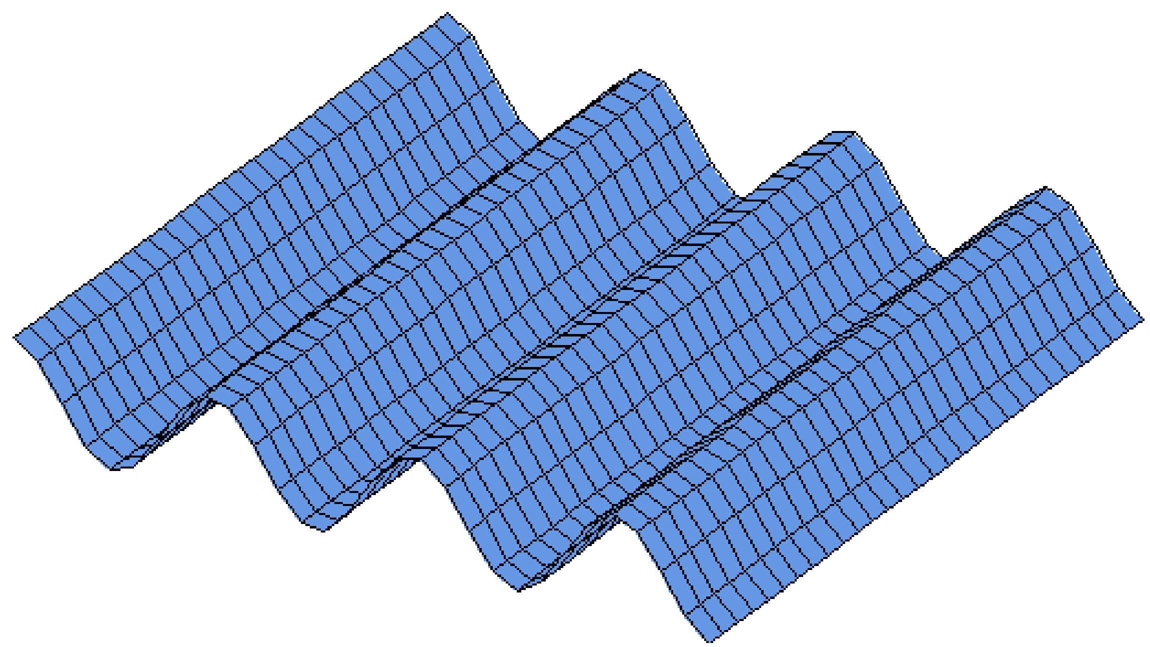
Só Arestas





Só FILL AREA

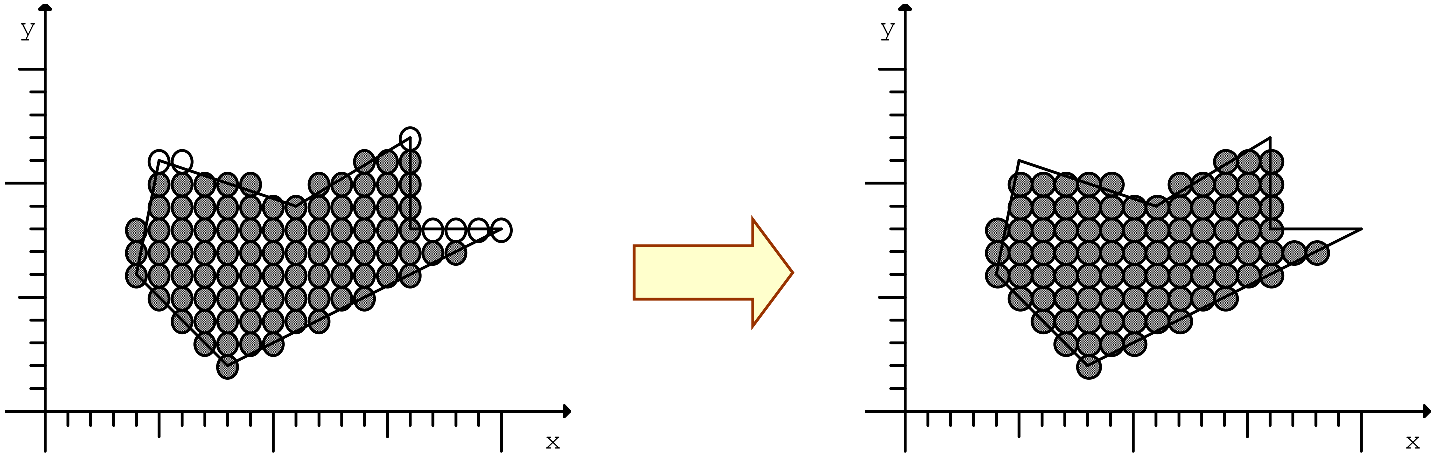
FILL AREA + Arestas



M.Próspero

Rasterização de Polígonos

Primeira correção do exemplo de referência, de acordo com o algoritmo dado:



M.Próspero

Problemas com a fronteira do preenchimento

- **Acumulação de erros por x+1/m nem sempre ser um inteiro.**

Resolução:

Trabalhar com os números na forma de fração.

Exemplos com arestas para vários valores de linhas de varrimento:

a3

m=5

y	6	7	8	9	10	11
x	4	4+1/5	4+2/5	4+3/5	4+4/5	5

a5

m=3/5

y	9	10	11	12
x	11	11+5/3 = 12+2/3	12+7/3 = 14+1/3	14+6/3 = 16

M.Próspero

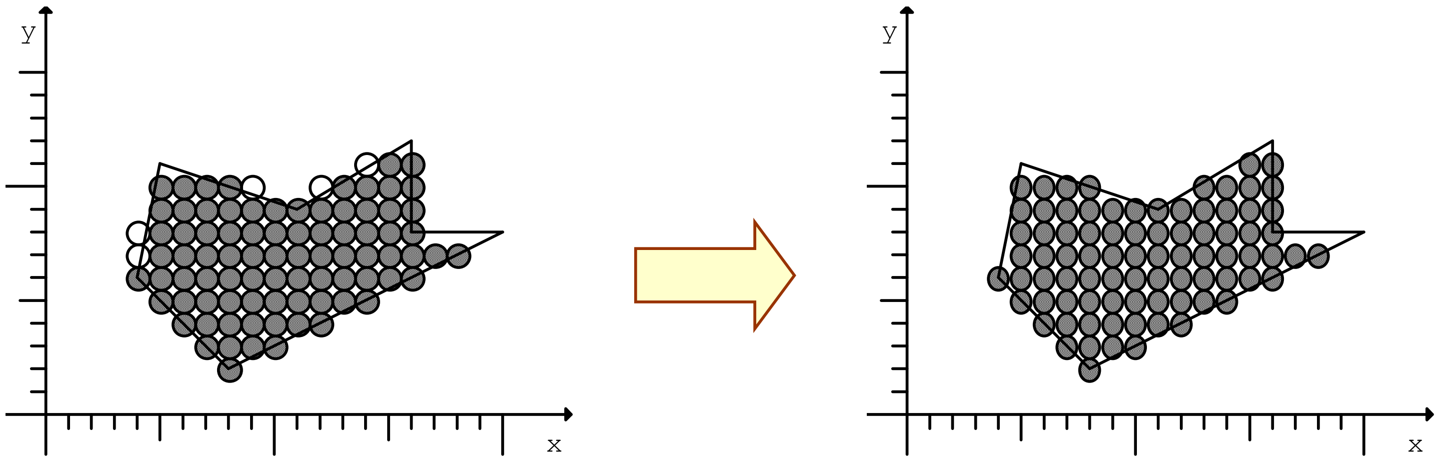
Problemas com a fronteira do preenchimento

- **Garantir que os limites das carreiras de pixels não são pontos exteriores ao polígono.**

Não sendo **x** inteiro e caminhando-se, na linha de varrimento, do…

… exterior para o interior do polígono (Ex.: `a3`) → arredondamento para cima

… interior para o exterior do polígono → truncatura



[Repare-se nos valores de **x** arredondados para **5** em **`a3`**, mas todas as intersecções em **`a1`** dão valores inteiros a **x**]

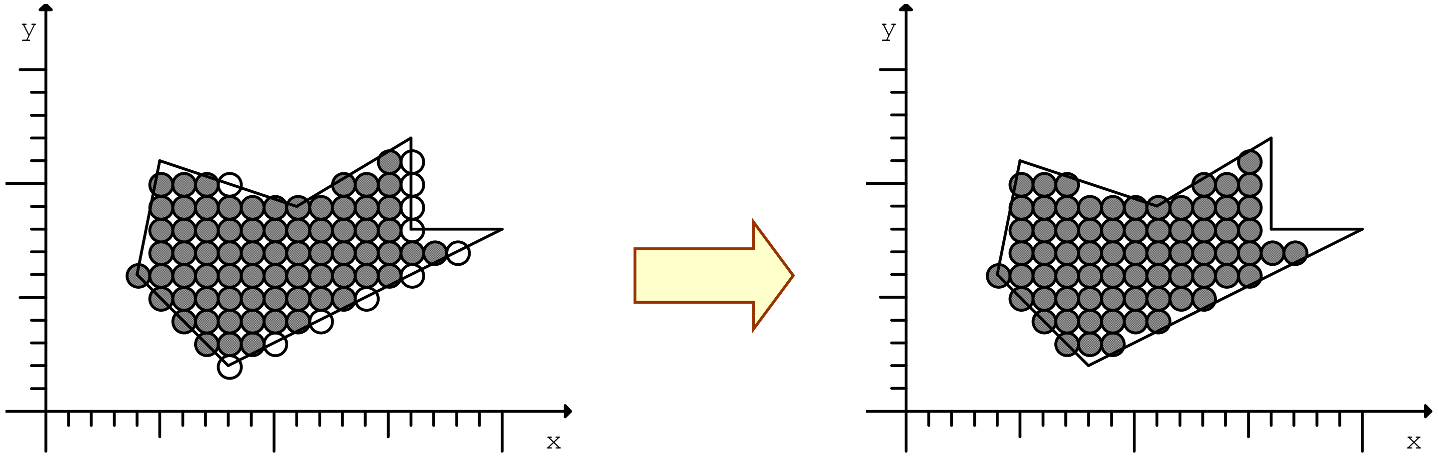
M.Próspero

Problemas com a fronteira do preenchimento

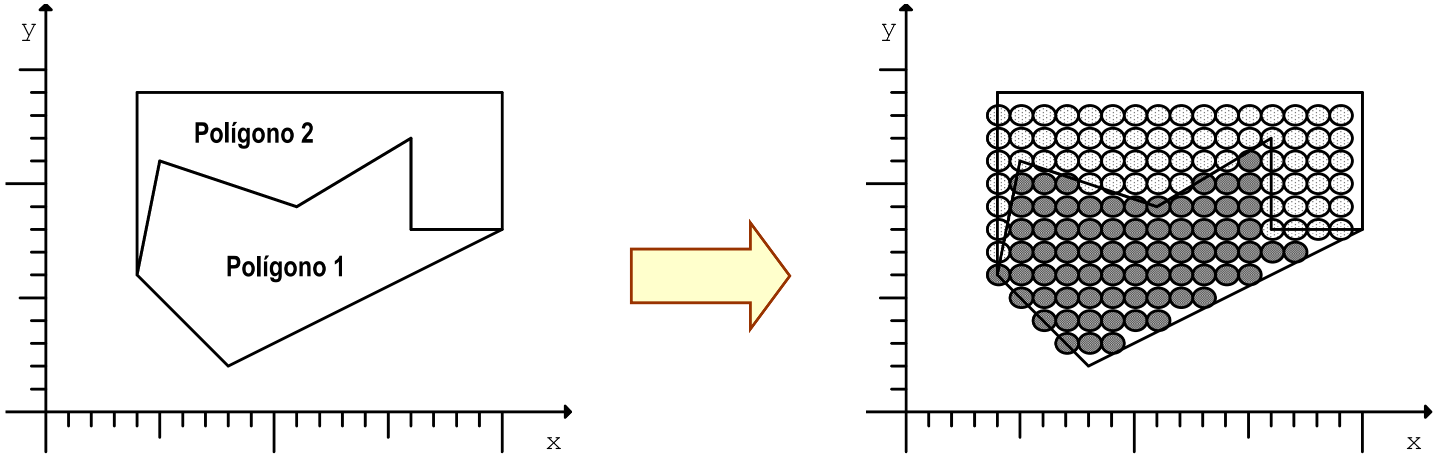
- **E se a intersecção der um valor inteiro para x ? É preciso evitar sobreposição com polígonos vizinhos!**

Uma convenção possível:
- Se é o pixel à esquerda na carreira (Ex.: **a2**) → o pixel é considerado 'interior'
- Se é o pixel à direita na carreira (Ex.: **a1**) → o pixel é considerado exterior

Prioritário em relação ao anterior



Aplicação: As malhas de polígonos são bem e completamente preenchidas, sem sobreposições.



M.Próspero