Aula 16

Preenchimento de Polígonos

Preenchimento de Polígonos

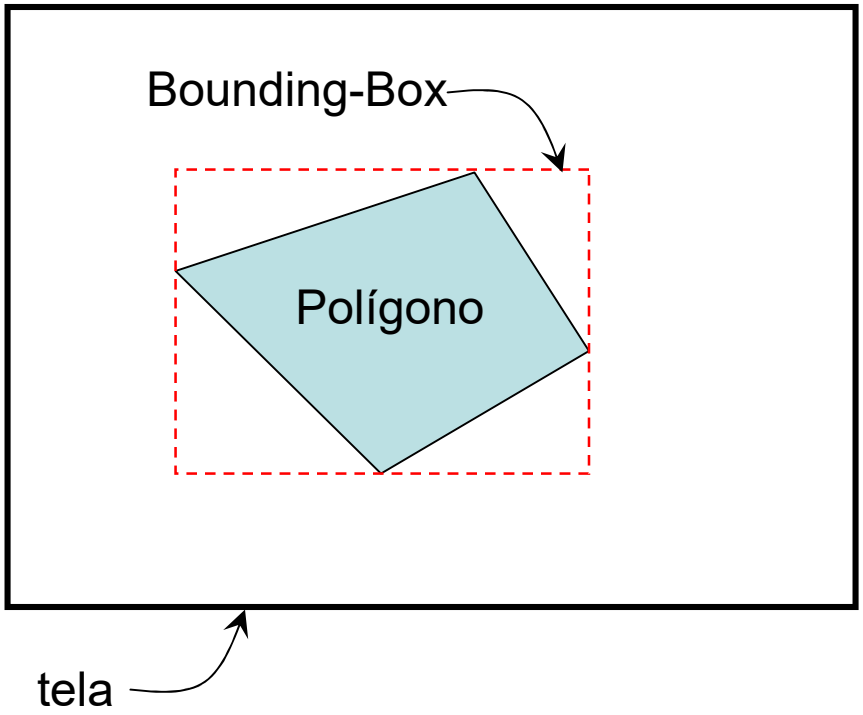
A maneira mais simples de preencher um polígono é testar se cada pixel da tela está dentro do polígono.

Porém, em geral, a maioria dos pixels não está dentro de um polígono qualquer, assim, esta é uma solução de pouca utilidade.

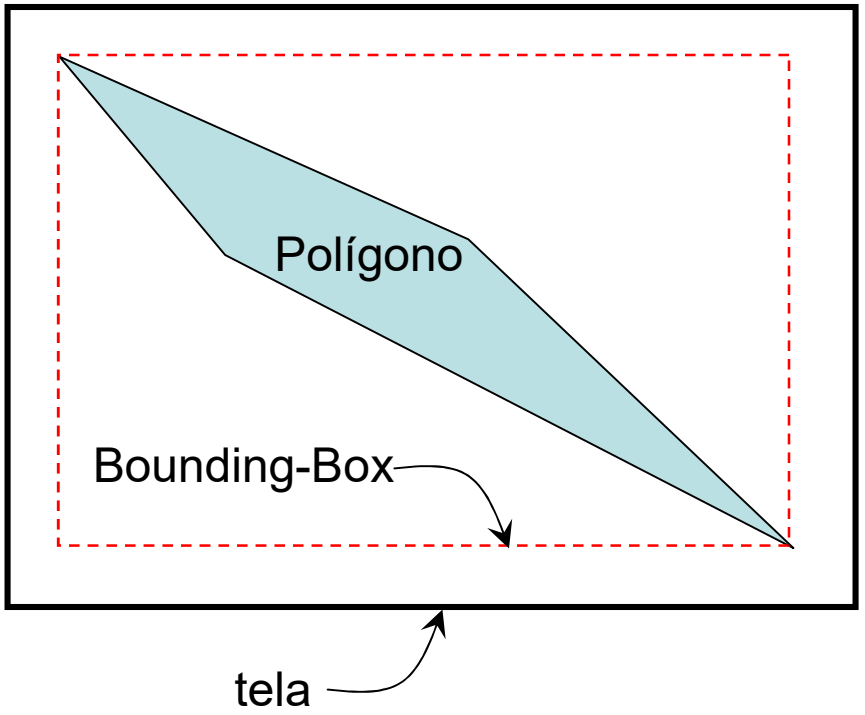
Seja usando esta técnica e também muitas outras, o uso de uma caixa mínima que contem o polígono, chamada **Bounding-Box** ajuda muito a reduzir o processamento

Preenchimento de Polígonos

Exemplos de uso do Bounding-Box



Grande Melhoria

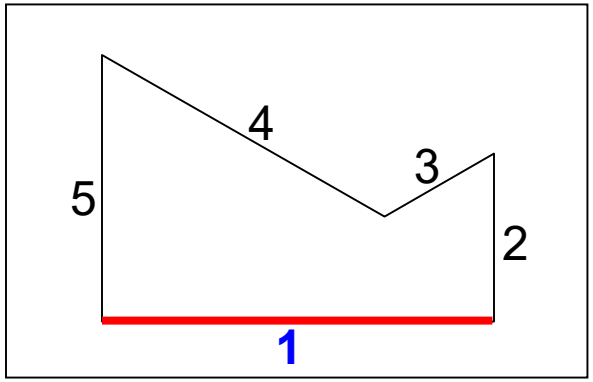


Pequena Melhoria

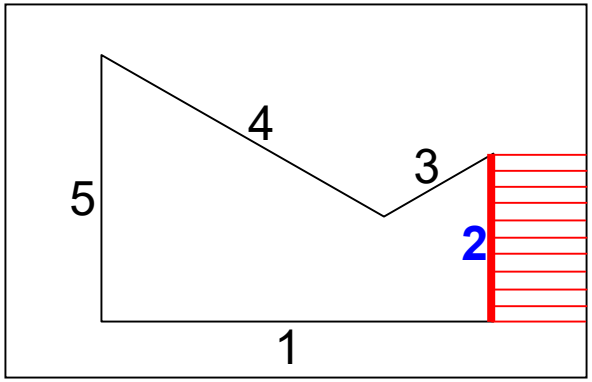
Preenchimento de Polígonos

Algoritmo de Inversão de Cores **(Edge Fill Algorithm)**

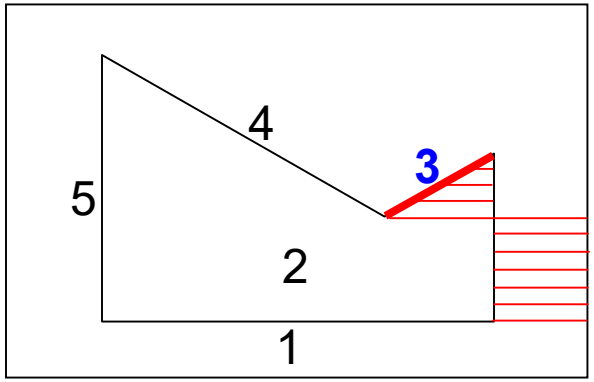
Trata-se de um algoritmo muito simples, que consiste em inverter todos os pixels à direita das poligonais (não horizontais) que delimitam o polígono.



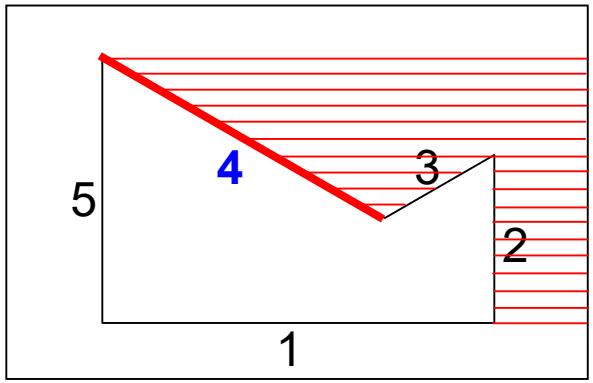
Poligonal 1



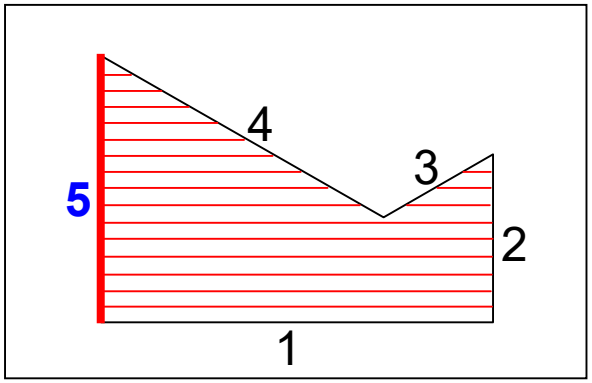
Poligonal 2



Poligonal 3



Poligonal 4

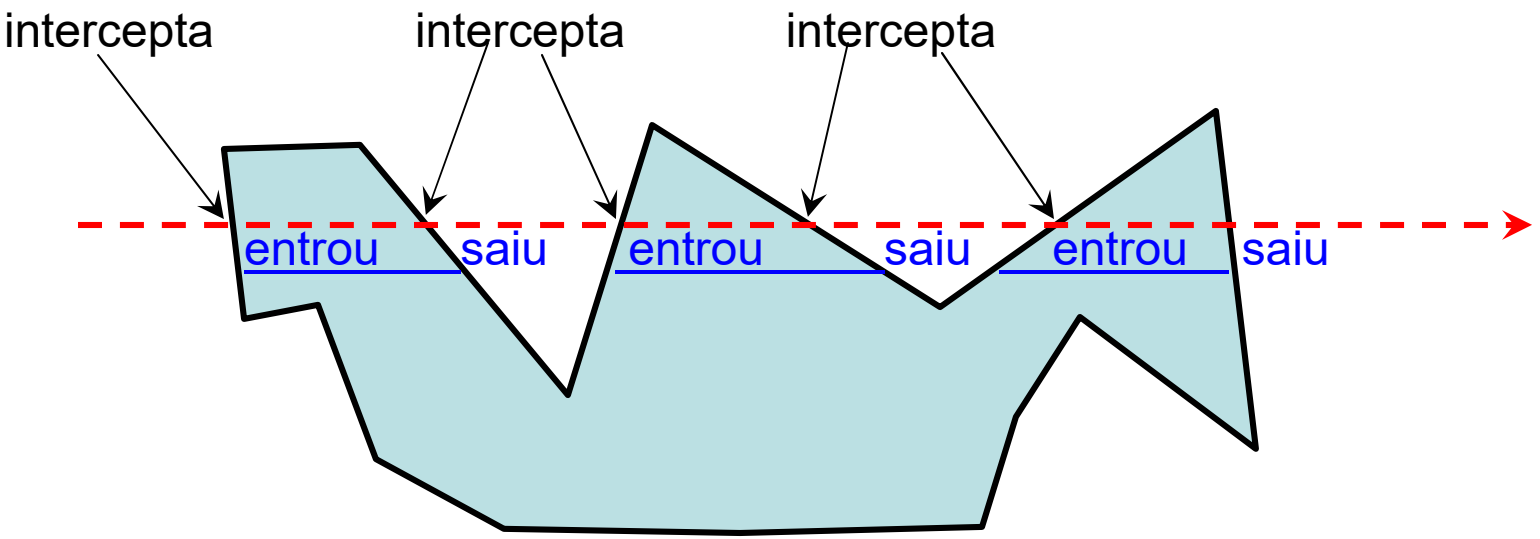


Poligonal 5

Preenchimento de Polígonos

Ordered List Algorithm (Scanline)

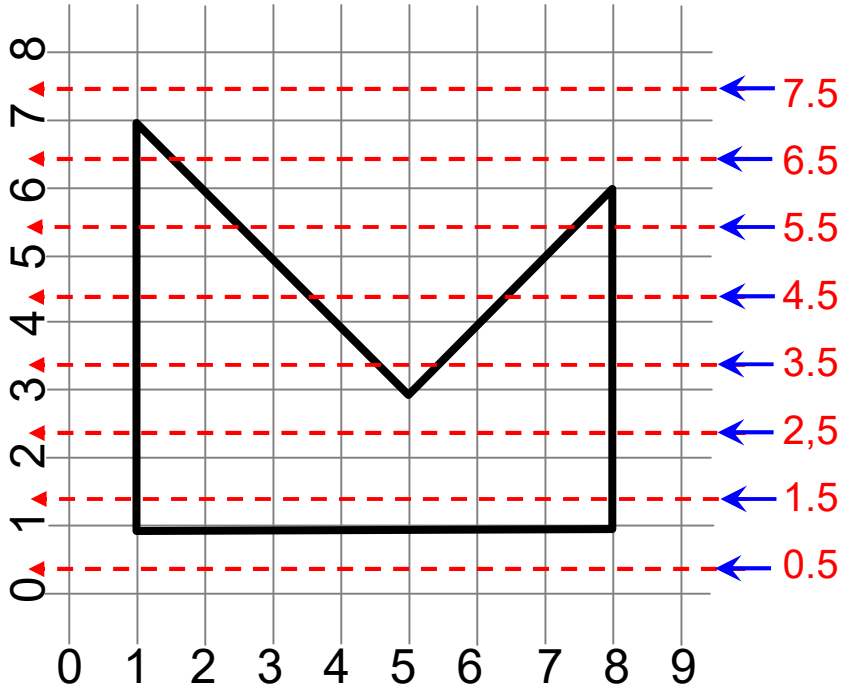
Usando a coerência espacial dos pixels, o preenchimento se dá, considerando que as características dos pixels em uma linha de varredura muda quando a linha intercepta uma polígonal do polígono



Preenchimento de Polígonos

Ordered List Algorithm (Scanline)

Determina as interseções com linhas de varredura e, em seguida, ordena a lista primeiro os maiores y_s e os menores x_s



1º) Usa linhas de varredura entre os pixels, da direita para a esquerda

linha 0.5: não intercepta

linha 1.5: (8, 1.5) (1, 1.5)

linha 2.5: (8, 2.5) (1, 2.5)

linha 3.5: (8, 3.5) (5.5, 3.5) (4.5, 3.5) (1, 3.5)

linha 4.5: (8, 4.5) (6.5, 4.5) (3.5, 4.5) (1, 4.5)

linha 5.5: (8, 5.5) (7.5, 5.5) (2.5, 5.5) (1, 5.5)

linha 6.5: (1.5, 6.5) (1, 6.5)

linha 7.5: não intercepta

Preenchimento de Polígonos

2º) Ordena a lista (primeiro os maiores y_s e menores x_s)

(1, 6.5) (1.5, 6.5)
(1, 5.5) (2.5, 5.5) (7.5, 5.5) (8, 5.5)
(1, 4.5) (3.5, 4.5) (6.5, 4.5) (8, 4.5)
(1, 3.5) (4.5, 3.5) (5.5, 3.5) (8, 3.5)
(1, 2.5) (8, 2.5)
(1, 1.5) (8, 1.5)

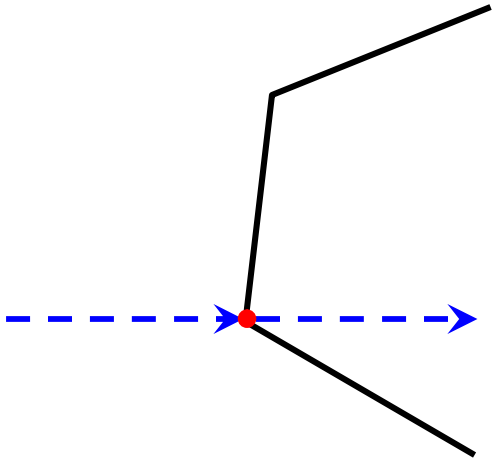
3º) Considerando a coerência espacial dos pixels, a lista de pixels a serem ligados é (considerando um truncamento)

(1, 6)
(1, 5) (2, 5) (7, 5) (8, 5)
(1, 4) (2, 4) (3, 4) (6, 4) (7, 4) (8, 4)
(1, 3) (2, 3) (3, 3) (4, 3) (5, 3) (6, 3) (7, 3) (8, 3)
(1, 2) (2, 2) (3, 2) (4, 2) (5, 2) (6, 2) (7, 2) (8, 2)
(1, 1) (2, 1) (3, 1) (4, 1) (5, 1) (6, 1) (7, 1) (8, 1)

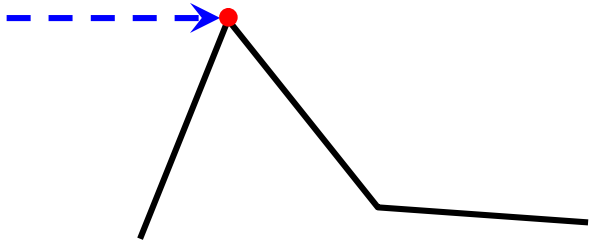
Preenchimento de Polígonos

É preciso cuidado extra quando uma linha de varredura intercepta um vértice do polígono, pois neste local, tem-se duas interseções da linha de varredura com as poligonais e, neste caso, pode ser preciso considerar uma ou duas interseções

ex.



Intercepta duas poligonais deve-se considerar apenas uma (entra no polígono)



Intercepta duas poligonais deve-se considerar duas (entra e sai do polígono)

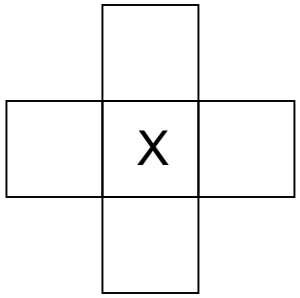
Preenchimento de Polígonos

Preenchimento por Inundação

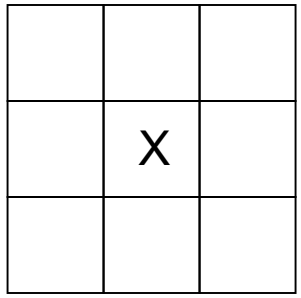
(Seed-Fill – Flood Fill - Boundary Algorithm)

A partir de um ponto no interior do polígono, ligue o ponto e chame o mesmo procedimento para os seus vizinhos, até atingir a borda do polígono

Pode-se usar a vizinhança 4 ou 8



Vizinhança 4



Vizinhança 8

Preenchimento de Polígonos

Preenchimento por Inundação

Algoritmo

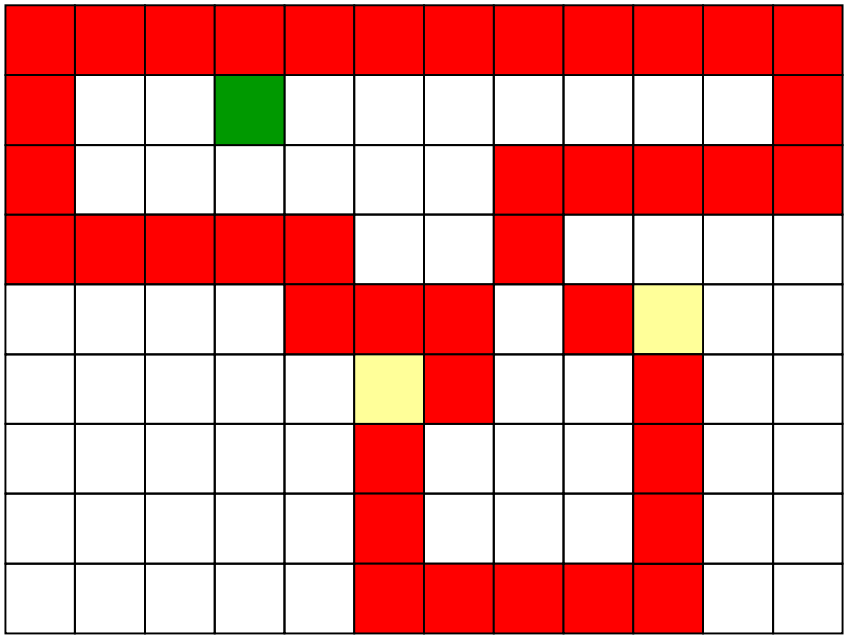
```
seed_fill(x,y,cor)
{
   pinta (x,y,cor)
   if  ( (x+1,y) ≠ borda) e ( (x+1,y) ≠ cor)
      seed_fill(x+1,y,cor)
   if  ( (x-1,y) ≠ borda) e ( (x-1,y) ≠ cor)
      seed_fill(x-1,y,cor)
   if  ( (x,y+1) ≠ borda) e ( (x,y+1) ≠ cor)
      seed_fill(x,y+1,cor)
   if  ( (x,y-1) ≠ borda) e ( (x,y-1) ≠ cor)
      seed_fill(x,y-1,cor)
}
```

Preenchimento de Polígonos

Prática - Implementar o preenchimento de polígonos usando:

1) Inversão de cores (usando o bouding-box)

2) Flood-fill usando a vizinhança 4 e 8



- Seed
- Borda
- Pontos que precisam ser ativos usando viz. 8