VISUALIZAÇÃO E RECORTE EM 2D

Computação Gráfica e Interfaces

FCT UNIVERSIDADE NOVA DE LISBOA

Sumário

Visualização em 2D

- Janela e Visor
- Enquadramento Janela-Visor

Recorte em 2D

- Algoritmos de Cohen-Sutherland e de Cyrus-Beck para recorte por janela rectangular
- Algoritmo de Sutherland-Hodgman para polígonos

Janela e Visor

Janela (*Window*)

- Área definida no sistema de coordenadas reais que delimita a informação a visualizar

Visor (*Viewport*)

- Área definida no sistema de coordenadas do dispositivo de saída (canónico ou efectivo), que delimita a zona onde será visualizada o conteúdo da janela

Visualização 2D

Recorte (*clipping*)

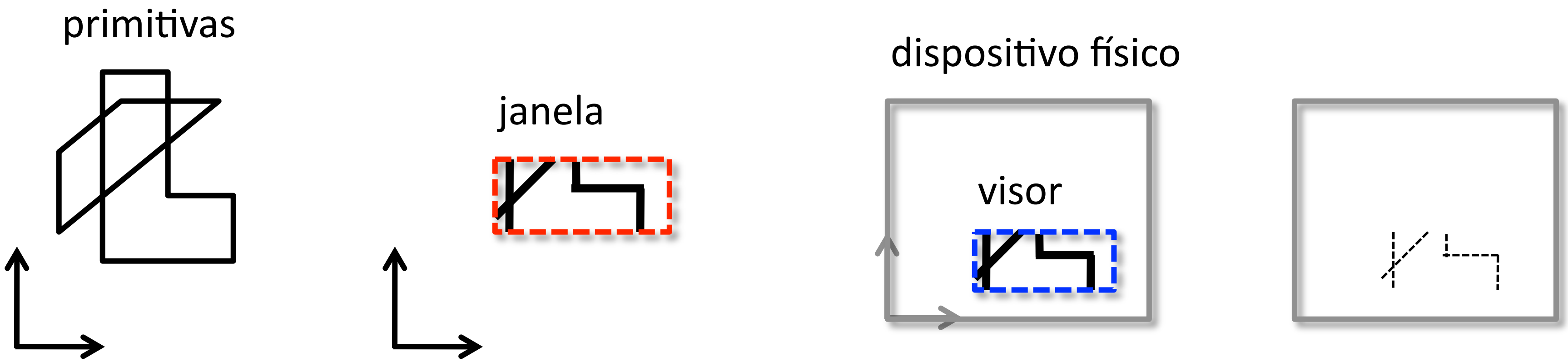
- Recorta/remove parte das primitivas geométricas exteriores à janela

Transformação de enquadramento (janela para visor)

- Transforma as primitivas já recortadas na janela para o visor

Conversão por varrimento

- Activa pixels do dispositivo físico, que vão representar as primitivas (e.g. numa impressora)

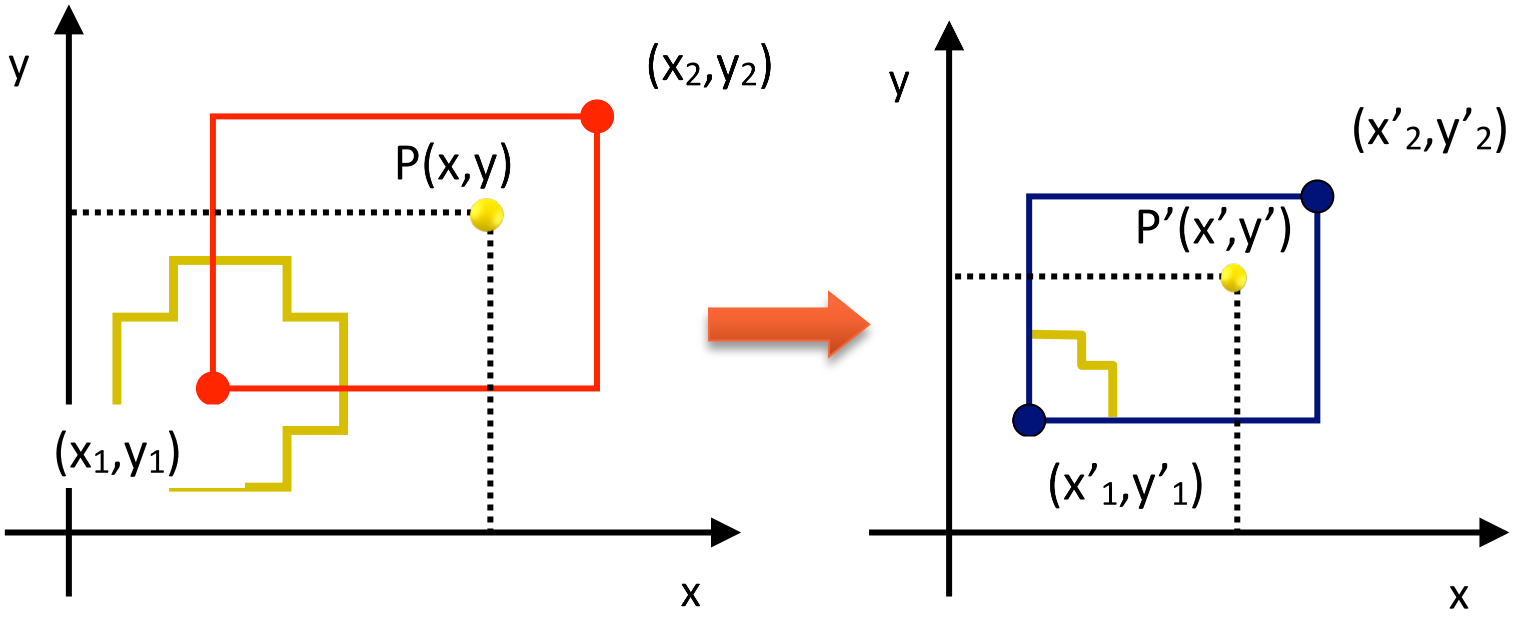


Coordenadas do utilizador
(World Coordinates ⇔ WC)

Coordenadas do dispositivo físico
(Device Coordinates ⇔ DC)

Transformação de enquadramento

Mapeamento janela-para-visor (*Window-to-viewport mapping*)

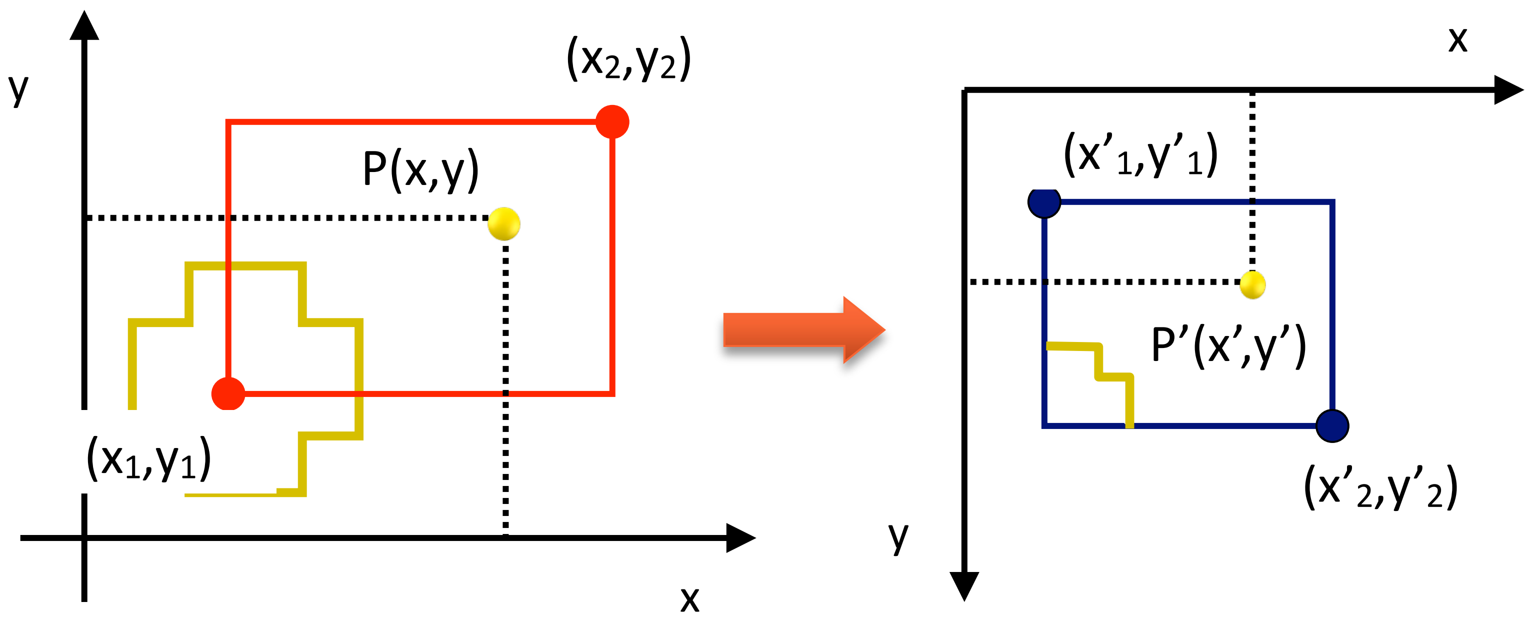


Janela, em coordenadas (x,y) do utilizador

Visor, em coordenadas (x', y') do dispositivo

$$\begin{cases} \dfrac{x_2 - x_1}{x - x_1} = \dfrac{x'_2 - x'_1}{x' - x'_1} \\[2ex] \dfrac{y_2 - y_1}{y - y_1} = \dfrac{y'_2 - y'_1}{y' - y'_1} \end{cases} \Rightarrow \begin{cases} x' = x'_1 + \dfrac{(x - x_1)(x'_2 - x'_1)}{(x_2 - x_1)} \\[2ex] y' = y'_1 + \dfrac{(y - y_1)(y'_2 - y'_1)}{(y_2 - y_1)} \end{cases}$$

Enquadramento para coordenadas de ecrã (origem no canto superior esquerdo)



Janela, em coordenadas (x,y) do utilizador

Visor, em coordenadas (x', y') do ecrã

$$\begin{cases} \dfrac{x_2 - x_1}{x - x_1} = \dfrac{x'_2 - x'_1}{x' - x'_1} \\ \dfrac{y_2 - y_1}{y - y_1} = \dfrac{y'_2 - y'_1}{y'_2 - y'} \end{cases} \Rightarrow \begin{cases} x' = x'_1 + \dfrac{(x - x_1)(x'_2 - x'_1)}{(x_2 - x_1)} \\ y' = y'_2 - \dfrac{(y - y_1)(y'_2 - y'_1)}{(y_2 - y_1)} \end{cases}$$

Relação de aspecto

Formato ou relação de aspecto (*aspect ratio*) de um rectângulo é a relação entre os lados:

$$a = \Delta x / \Delta y$$

Alguns exemplos comuns:

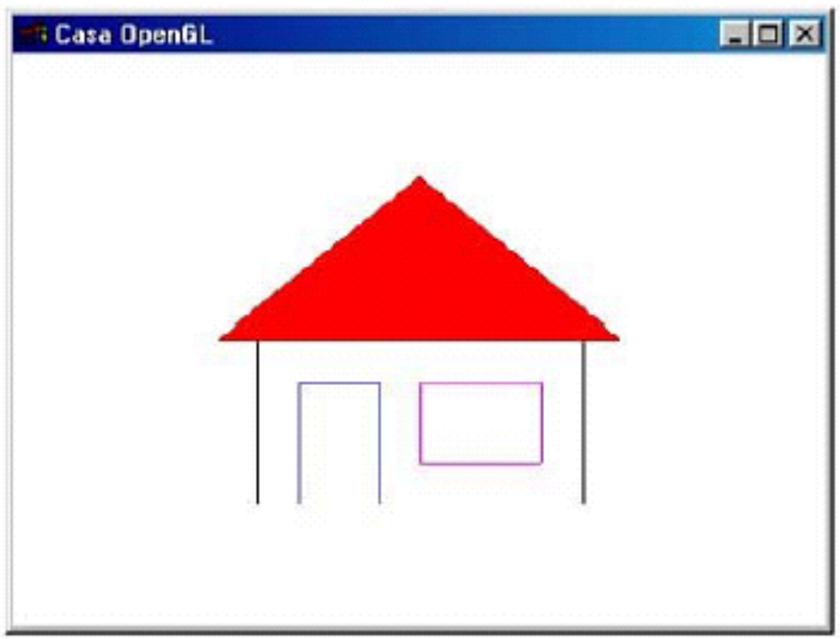
4:3 — Formato vídeo standard

16:9 — Televisão de alta-definição

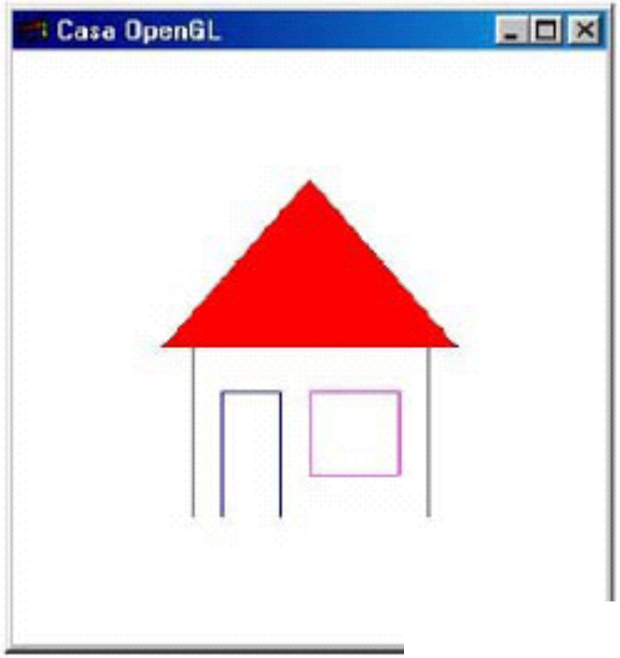
Distorção de imagem

Programa de demonstração

Quando a janela e o visor não têm a mesma relação de aspecto, a imagem fica distorcida



janela (0 , 400 , -280 , 0)
visor (0 , 400 , 0 , 280)
a = 10 : 7



janela (0 , 400 , -280 , 0)
visor (0 , 280 , 0 , 280)
$a_v = 1 : 1$

Imagens: M. Próspero dos Santos

$a_{janela} = 400/280 = 10/7$
$a_{visor} = 400/280 = 10/7$

$a_{janela} = 400/280 = 10/7$
$a_{visor} = 280/280 = 1$

Sumário

Visualização em 2D

- Janela e Visor
- Enquadramento Janela-Visor

Recorte em 2D

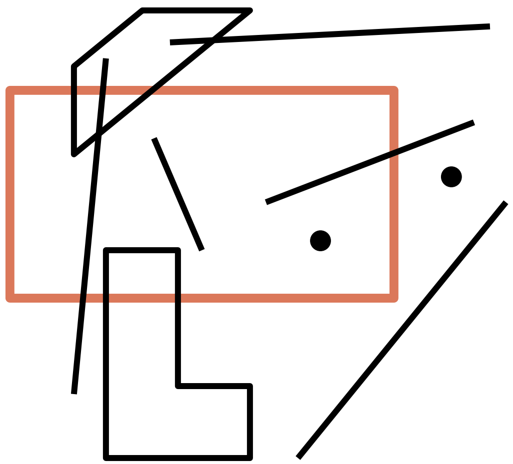
- Algoritmos de Cohen-Sutherland e de Cyrus-Beck para recorte por janela rectangular
- Algoritmo de Sutherland-Hodgman para polígonos

Recorte por janela rectangular

Motivação

Não devemos perder tempo e esforço com primitivas que, no todo ou em parte, se localizam fora da janela de visualização, ou seja, da janela de recorte. O utilizador apenas visualizará informação contida na janela.

Para a extracção do que se pretende visualizar, por vezes é removida grande parte das primitivas. Assim, esta operação de pré-processamento é bastante importante, devendo ser rápida e se possível com implementação em hardware

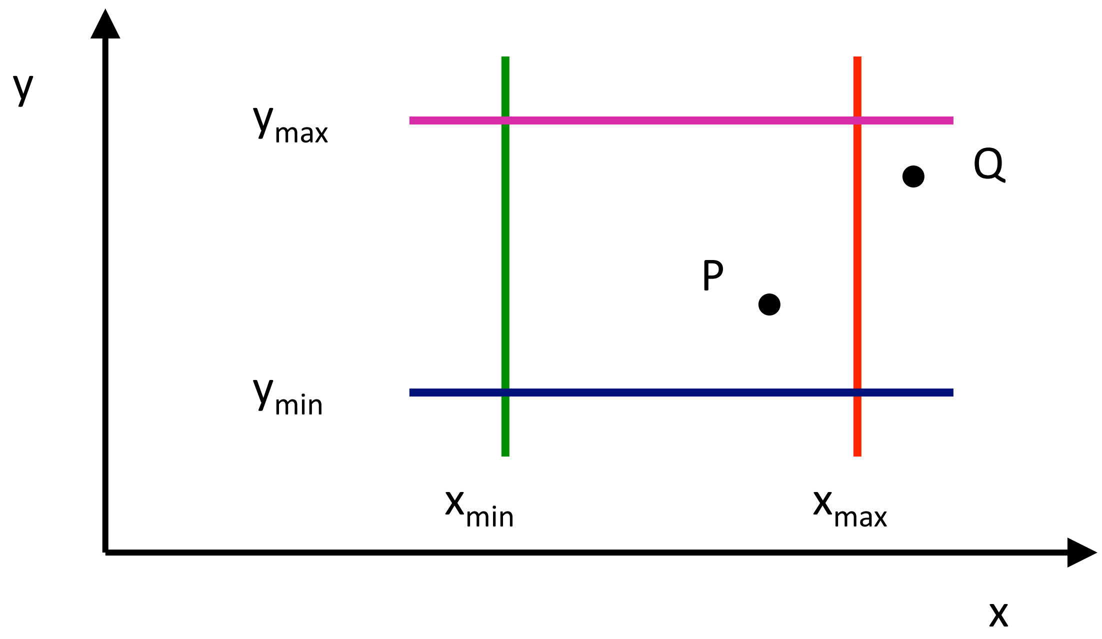


Recorte de pontos por janela rectangular

P(x,y) é visível se

$$x_{min} <= x <= x_{max}$$

$$y_{min} <= y <= y_{max}$$



Algoritmo de Cohen-Sutherland

Princípios

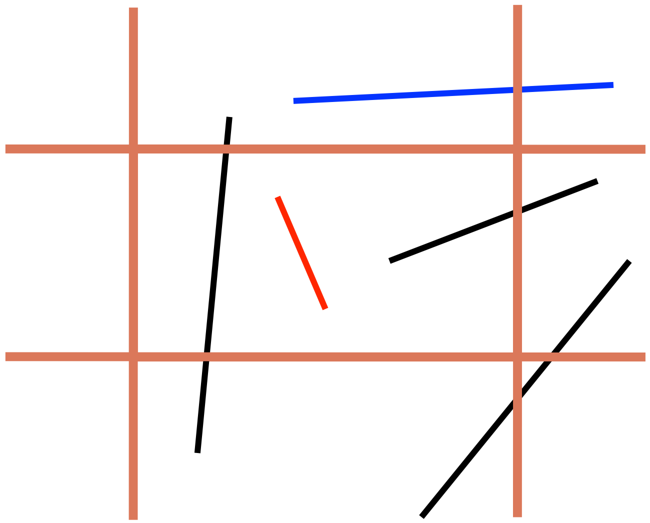
- Segmento de recta com codificação dos pontos extremos relativamente aos limites da janela
- Dois níveis de decisão: imediata (trivial) ou intermédia

Decisão imediata

- Através da análise dos respectivos códigos, é possível afirmar que o segmento está fora da janela, ou então, que está dentro da janela,

Decisão intermédia

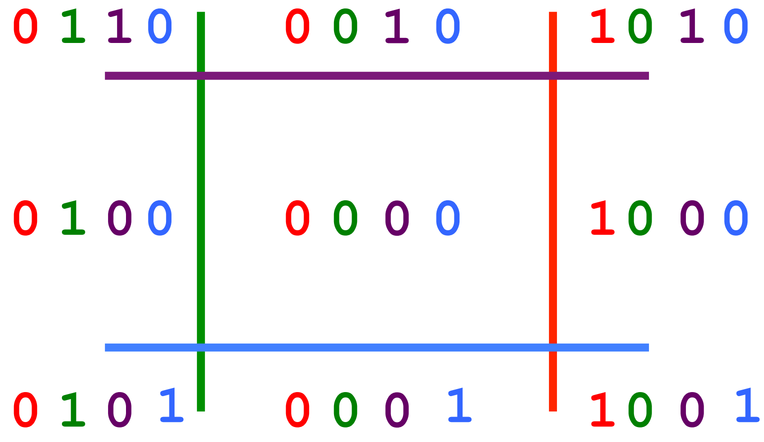
- Não havendo conclusão trivial, subdivide-se o segmento e o processo é repetido



1. Aceita imediatamente se ambos os extremos são interiores a todos os limites da janela
2. Rejeita imediatamente se ambos os extremos são exteriores ao mesmo limite da janela

Codificação

- Definição de regiões de teste em relação à janela rectangular e atribuição de um código binário (4 bits) a cada extremo da linha, em função da sua posição relativamente à janela



dir	esq	cim	bai

Com a ordem indicada (que poderia ser escolhida outra)

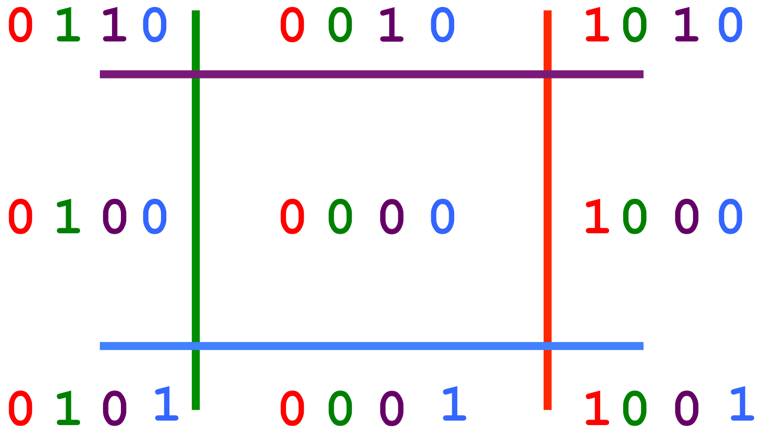
bit 1 : 1, se $x > x_{max}$

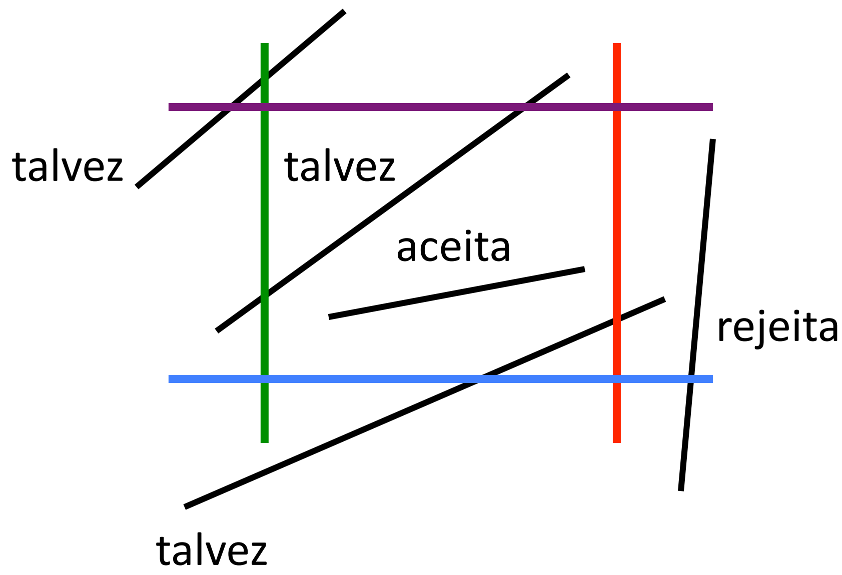
bit 2 : 1, se $x < x_{min}$

bit 3 : 1, se $y > y_{max}$

bit 4 : 1, se $y < y_{min}$

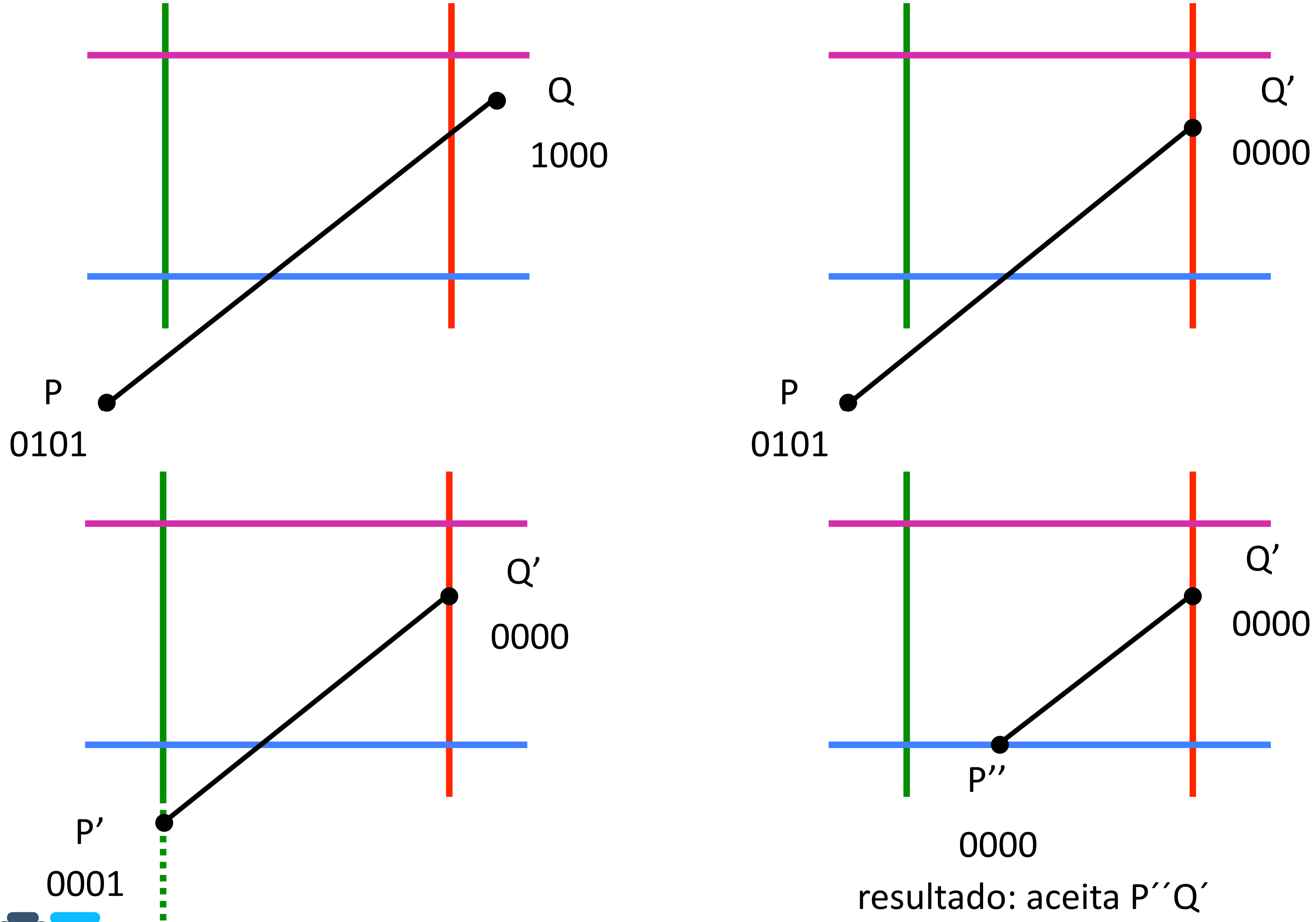
```
if (code1==0 && code2==0)
{ "trivial:
    segmento dentro … aceita"
}
else if (code1 & code2 != 0)
{ "trivial:
   segmento fora … rejeita"
}
else
{ "… talvez:
    subdivide o segmento e
    repete o processo"
}
```





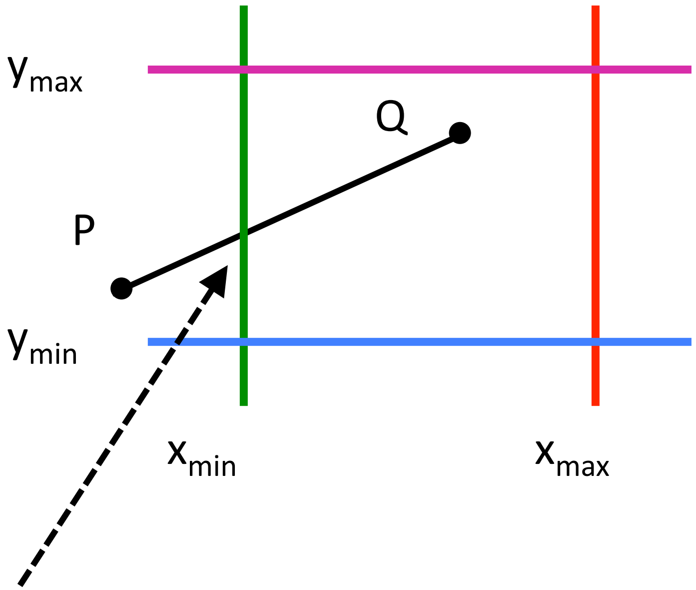
Exemplo

Ordem de codificação:	dir	esq	cim	bai
Ordem de processamento dos limites da janela:	dir	esq	cim	bai



Intersecção com os limites da janela

- Utilização da equação paramétrica do segmento
- Na iteração seguinte, o ponto de intersecção vai substituir um ponto que de momento se situa no exterior da janela



$$\begin{cases} x = x_P + t\,(x_Q - x_P) \\ y = y_P + t\,(y_Q - y_P) \end{cases}$$

Exemplo do cálculo da intersecção de PQ com a aresta $x = x_{min}$

$t = 0 \Rightarrow P$

$t = 1 \Rightarrow Q$

$$t = \frac{x_{min} - x_P}{x_Q - x_P}$$

$$\begin{cases} x = x_{min} \\ y = y_P + \dfrac{x_{min} - x_P}{x_Q - x_P}\,(y_Q - y_P) \end{cases}$$

Como seria em 3D ?

Programa de demonstração

Algoritmo de Cyrus-Beck

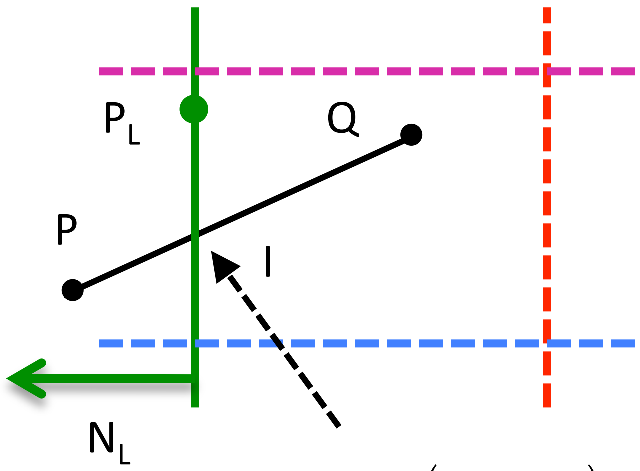
Princípios

- Optimização da intersecção do segmento de recta com os limites da janelas utilizando a formulação paramétrica
- Recorre-se também a um ponto em cada aresta da janela e respectivas normais
- Determinação dos valores de t para os quais o segmento entre e sai da janela, ou concluir se o segmento está totalmente fora da janela

$$P(t) = P + t\,(Q - P)\,,\ \text{com}\ 0 \le t \le 1$$

Metodologia para cada um dos limites da janela

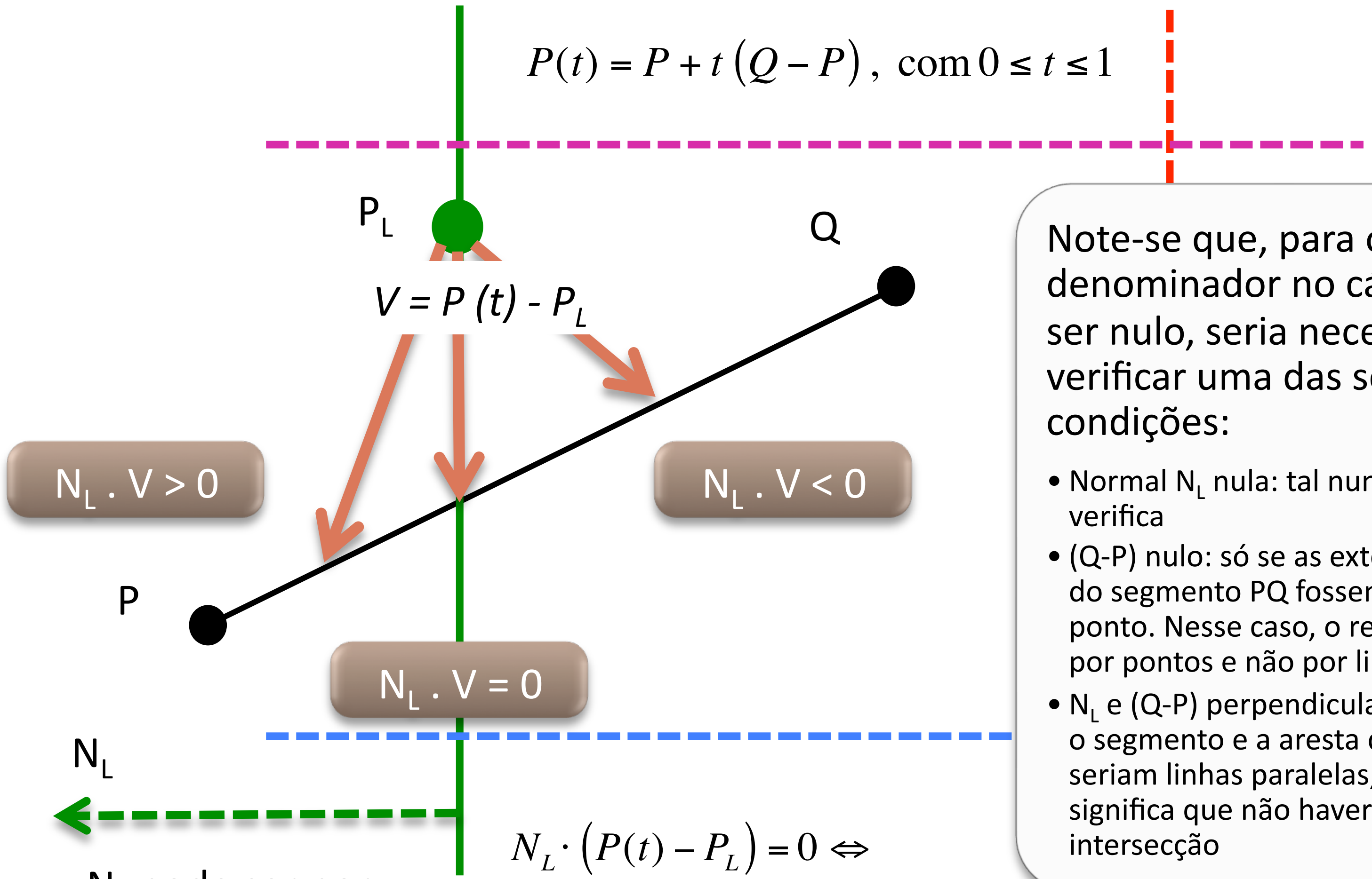
- Exemplificando com o limite esquerdo da janela, a partir da equação paramétrica do segmento de recta PQ, de um ponto situado nessa aresta da janela, P_L, e da normal a essa aresta, dirigida para o semi-espaço exterior à janela, N_L, o objectivo é determinar t tal que este corresponda ao ponto de intersecção I, ou seja, se for tal que o produto interno de N com P(t)-PL seja nulo



$$t\ :\quad N_L \cdot \left(P(t) - P_L\right) = 0$$

Obs.: Apresentamos o algoritmo na sua versão optimizada, isto é, segundo a formulação de Liang-Barsky (decorrente do uso de janela rectangular)

$$P(t) = P + t\,(Q - P)\,,\ \text{com}\ 0 \le t \le 1$$



N_L pode ser por exemplo (-1,0)

Note-se que, para o denominador no cálculo de t ser nulo, seria necessário verificar uma das seguintes condições:

- Normal N_L nula: tal nunca se verifica
- (Q-P) nulo: só se as extermidades do segmento PQ fossem o mesmo ponto. Nesse caso, o recorte seria por pontos e não por linhas
- N_L e (Q-P) perpendiculares entre si: o segmento e a aresta da janela seriam linhas paralelas, o que significa que não haveria intersecção

$$N_L \cdot \left(P(t) - P_L\right) = 0 \Leftrightarrow$$

$$N_L \cdot \left(P + t\,(Q - P) - P_L\right) = 0 \Leftrightarrow$$

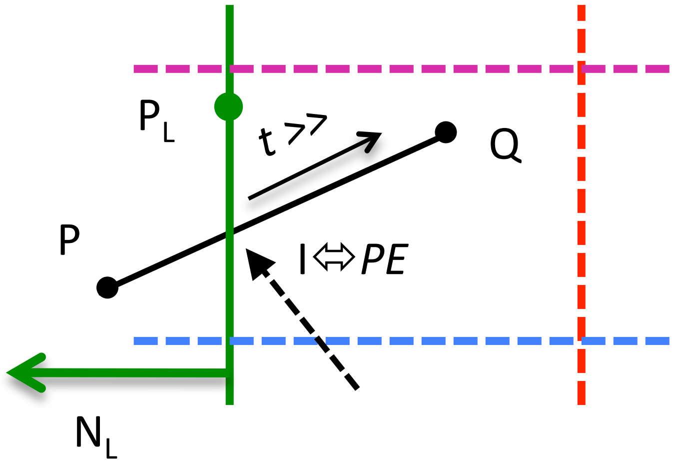
$$N_L \cdot \left(P - P_L + t\,(Q - P)\right) = 0 \Leftrightarrow$$

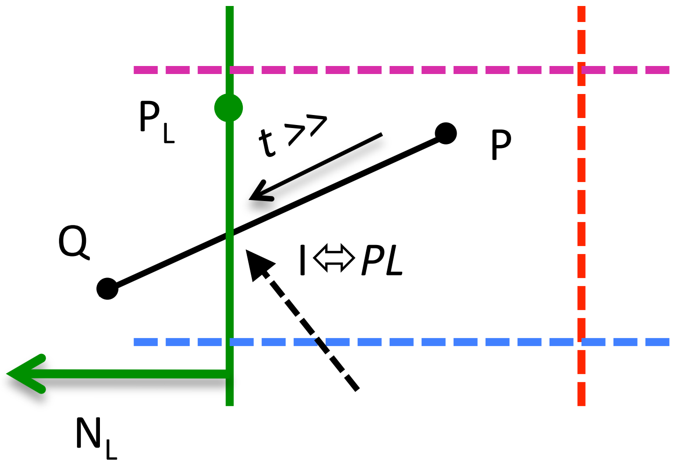
$$N_L \cdot \left(P - P_L\right) + N_L \cdot t\,(Q - P) = 0 \Leftrightarrow$$

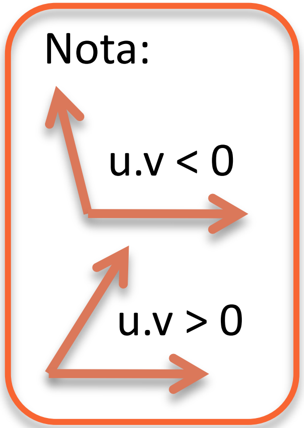
$$t = \frac{N_L \cdot \left(P - P_L\right)}{-N_L \cdot (Q - P)}\ \ , \text{com}\ \ N_L \cdot (Q - P) \neq 0$$

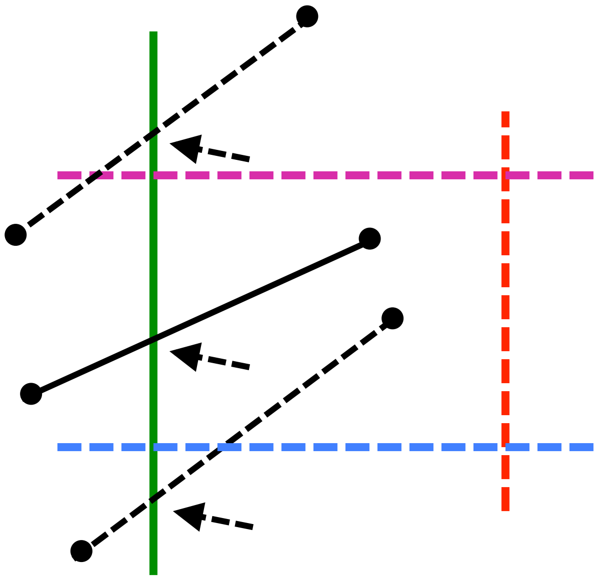
$$N_L \cdot \left(P(t) - P_L\right) = 0 \Leftrightarrow$$

$$t = \frac{N_L \cdot \left(P - P_L\right)}{-N_L \cdot \left(Q - P\right)} \quad \text{, com} \quad N_L \cdot \left(Q - P\right) \neq 0$$









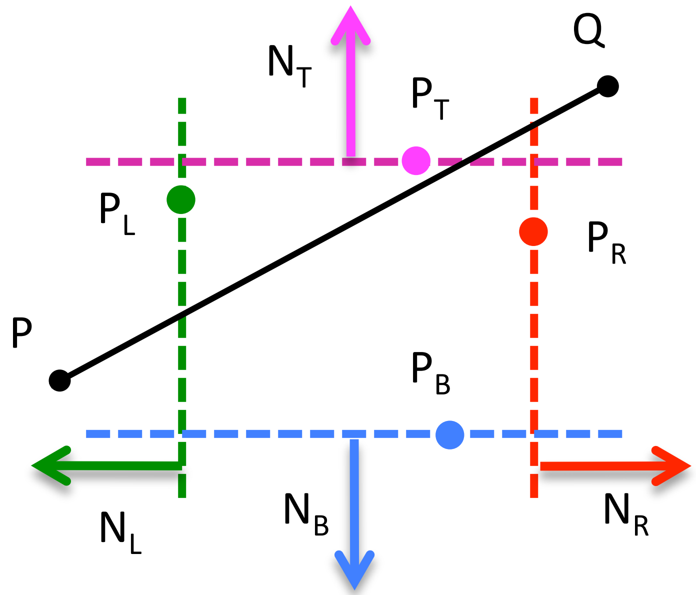
O cálculo de intersecção com o limite esquerdo da janela pode não corresponder a uma intersecção com a janela

- Se $N_L.(Q\text{-}P) < 0$, o ponto de intersecção PE é um ponto potencialmente de entrada na janela (Potentially Entering)
- Se $N_L.(Q\text{-}P) > 0$, o ponto de intersecção PL é um ponto potencialmente de saída da janela (Potentially Leaving)

Obs. A classificação do ponto de intersecção re-utiliza o cálculo do parâmetro t

A metodologia aplica-se a todas as arestas da janela

- No caso da janela rectangular {Bottom, Left, Top, Right}, utilizam-se respectivamente as quatro normais (0, -1), (-1, 0), (0, 1), (1, 0) e os pontos de escolha nas quatro arestas são constantes

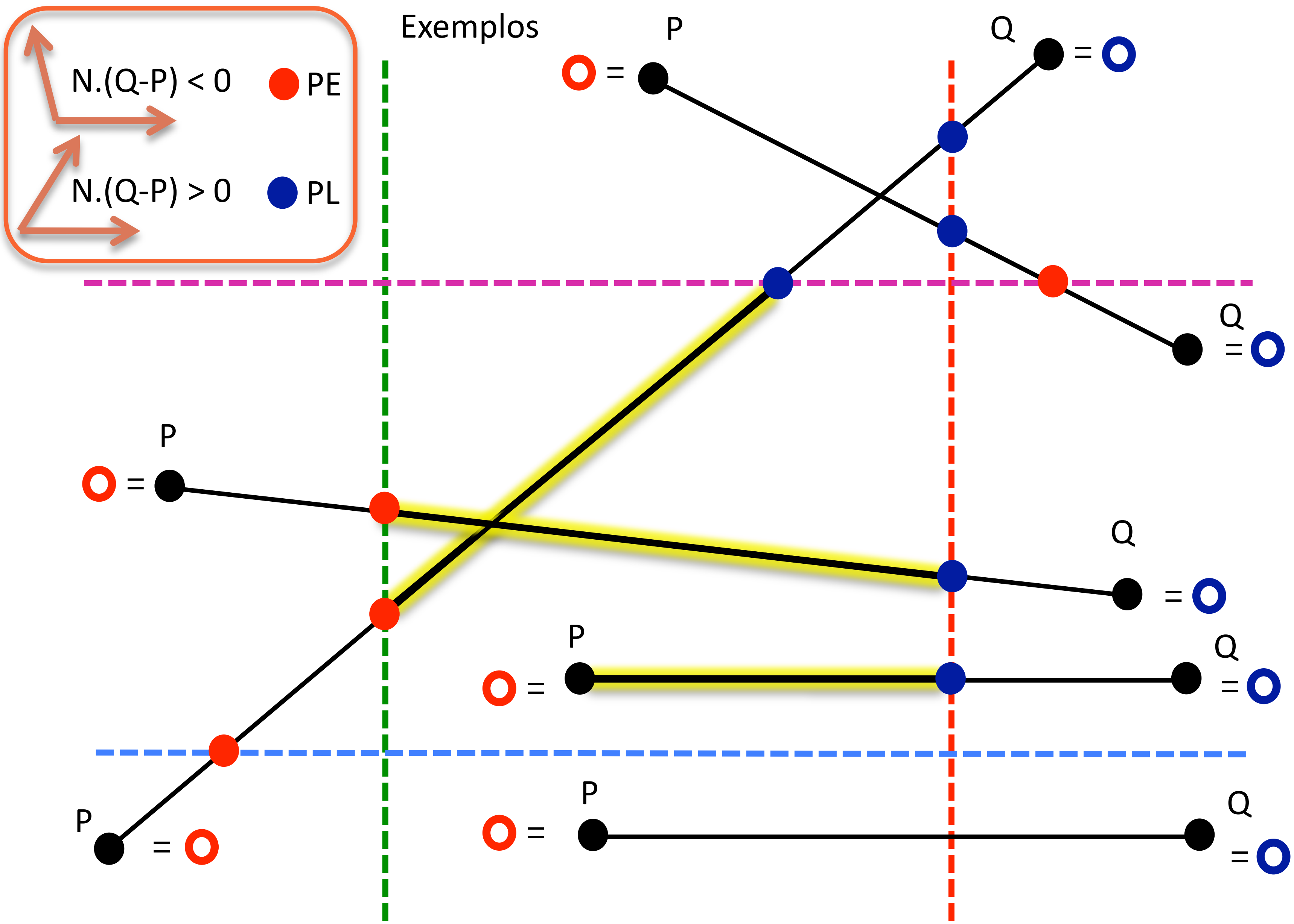


Dado o segmento PQ na sua forma paramétrica, o algoritmo determina os 4 valores de t e só posteriormente é que infere quais são as verdadeiras intersecções (zero ou duas)

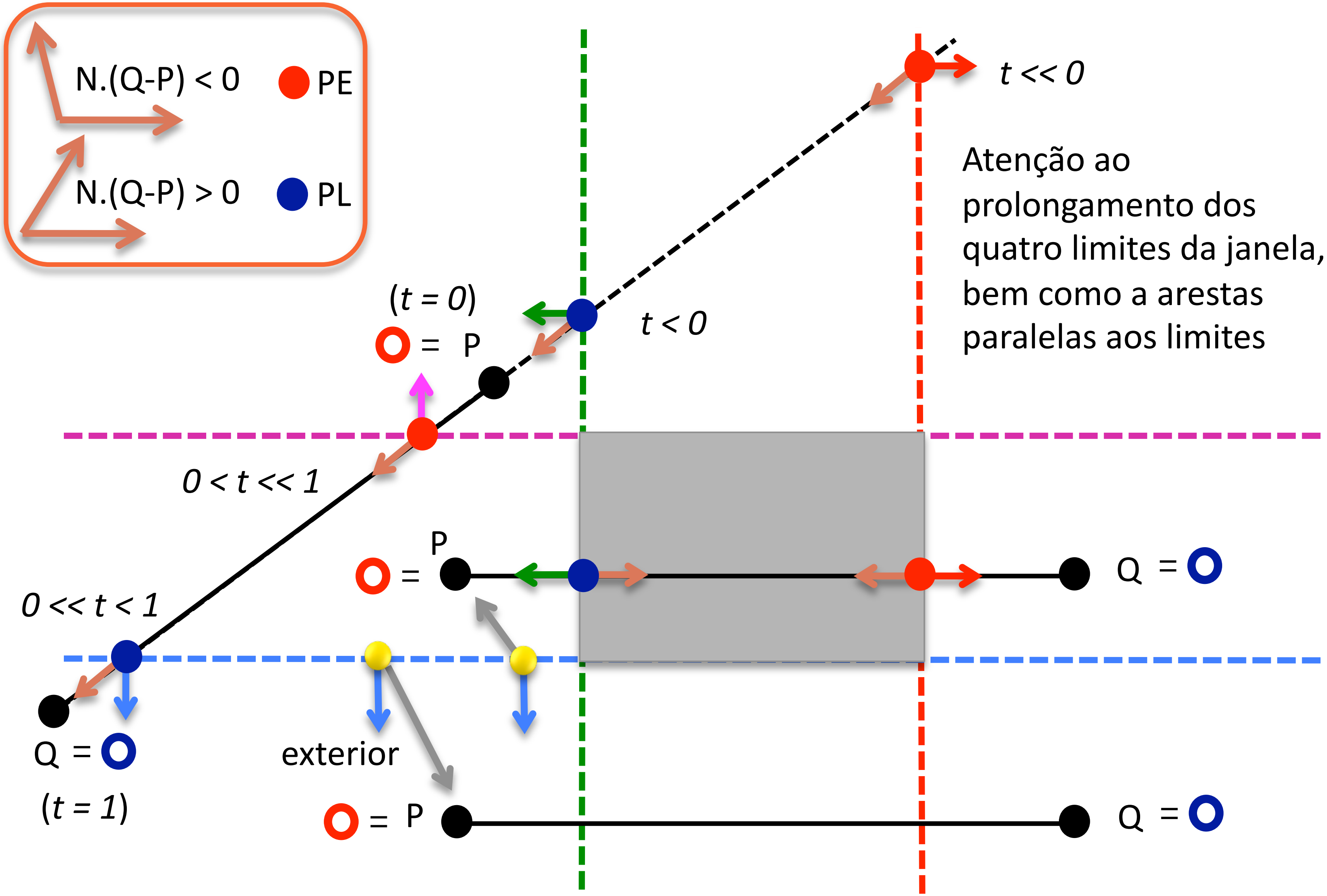
- Um ponto não é de intersecção se $t \notin [0,1]$
- Cada ponto de intersecção válido é classificado como PE (potentially entering) ou PL (potentially leaving)
- O menor par (PE, PL) define o segmento recortado. Assim, escolhe-se o maior t para PE (inicialmente a 0) e o menor t para PL (inicialmente a 1).
- No final, se o t do PE for maior do que o t do PL, o segmento é descartado. Senão, determinam-se as extremidades do segmento recortado através da utilização dos valores de t na equação paramétrica do segmento PQ

Algoritmo de Cyrus-Beck para recorte de segmentos de recta

```
Compute Ni e select Pi for each edge
for (each line PQ to be clipped)
   if (P = Q) then result is a point
   else
      tE = 0; tL = 1;
      for (each candidate intersection with a clip edge i)
        if (Ni.(Q-P) <> 0) then
           { note: parallel edges to line are ignored BUT
              if PQ is parallel to clip edge i and is in
              the exterior then result is automatically nil }
           compute t of line and clip edge intersection
           if (Ni.(Q-P) < 0) then tE = max(tE,t)  { PE }
           else tL = min(tL,t)  { PL }
         endif
      endfor
      if (tE > tL) then result is nil
      else result is P(tE) and P(tL) as true clip intersections
endfor
```



Ordem utilizada: Baixo, esquerda, direita, cima $P(t) = P + t\left(Q - P\right),\ \text{com}\, 0 \le t \le 1$



Atenção ao prolongamento dos quatro limites da janela, bem como a arestas paralelas aos limites

$$P(t) = P + t\left(Q - P\right),\ \text{com}\ 0 \le t \le 1$$

Comparação entre algoritmos de recorte

Algoritmo de Cohen-Sutherland

- Operação repetitiva de recorte é relativamente pesada
- Apropriado para quando houver vários segmentos fora da janela rectangular; assim pode aceitar ou rejeitar trivialmente

Algoritmo de Cyrus-Beck

- Cálculo das intersecções em t é simples
- Pontos de recorte só são determinados uma vez
- Não considera as situações de aceitação/rejeição triviais
- Apropriado para quando existem muitos segmentos para recortar
- Adaptável a quaisquer janelas poligonais convexas

Obs.: Ambos os algoritmos de Cohen-Sutherland e Cyrus-Beck/Liang Barsky são adaptáveis para 3D

Recorte de polígonos

Princípio geral

- Cada aresta da janela é considerada individualmente
- Recorte do polígono pela aresta da janela
- O polígono fica totalmente recortado após o recorte por todas as arestas da janela

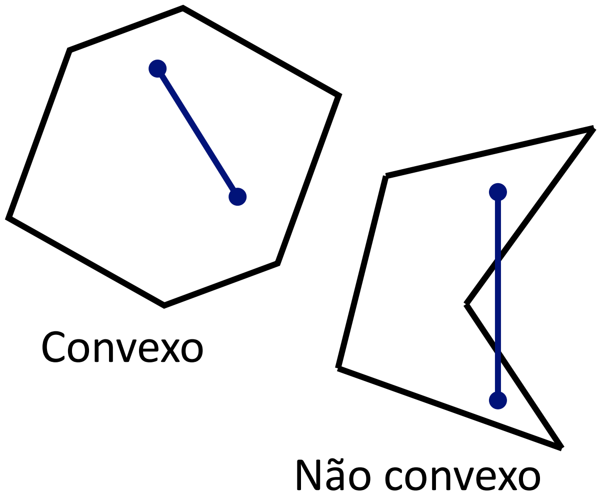
Algoritmo

- Entrada: lista ordenada dos vértices do polígono
- Saída: lista de vértices do polígono recortado, podendo ser vértices iniciais ou novos

Alguns conceitos sobre polígonos

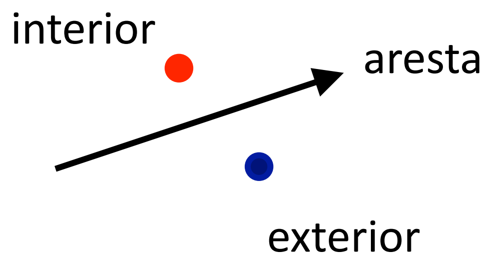
Convexidade

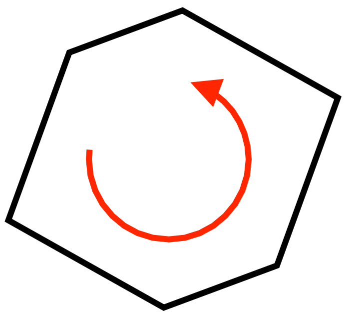
- Um polígono diz-se convexo se um segmento de recta unindo quaisquer dois pontos interiores estiver totalmente no interior



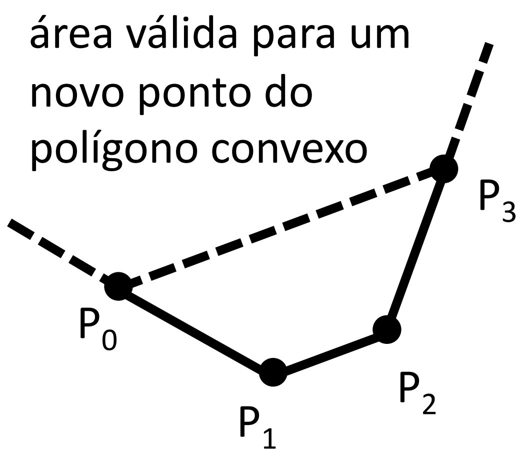
Ponto interior

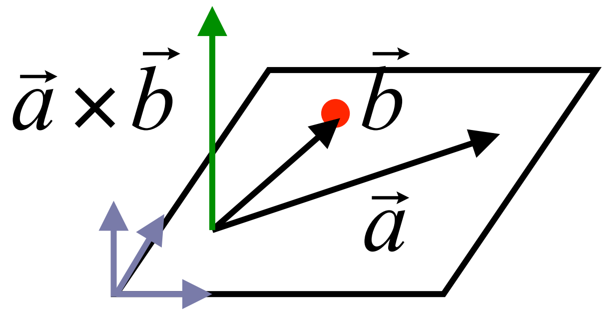
- Recorre-se ao sinal do produto externo entre vectores do plano - mesma origem, um com direcção da aresta e o outro dirigido ao ponto a testar



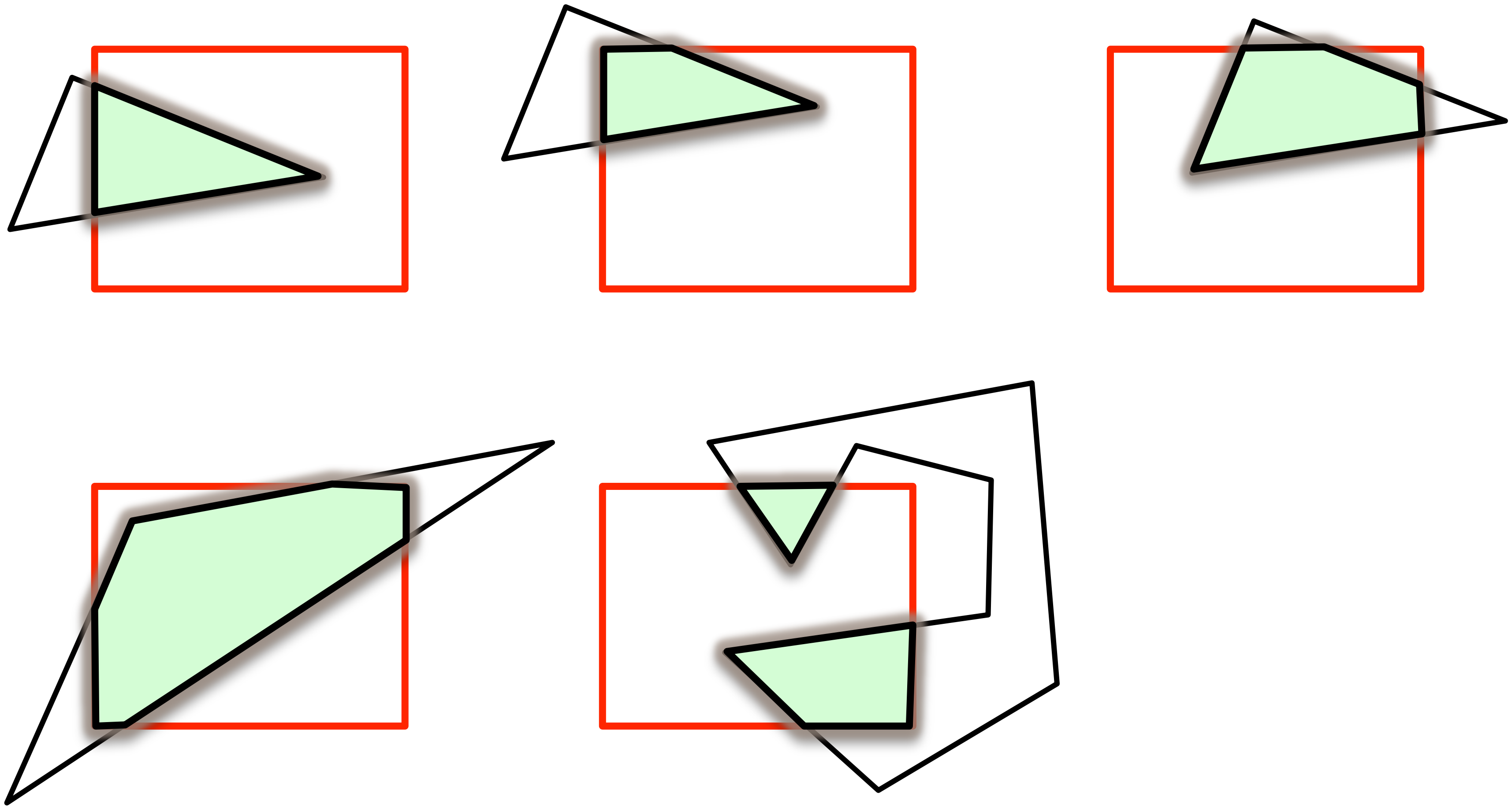


Orientação positiva das arestas





Exemplos



Polígono	Resultado do recorte
Triângulo	Polígono com 3, 4, 5 ou 7 lados
Não convexo	Vários polígonos

Algoritmo de Sutherland-Hodgman

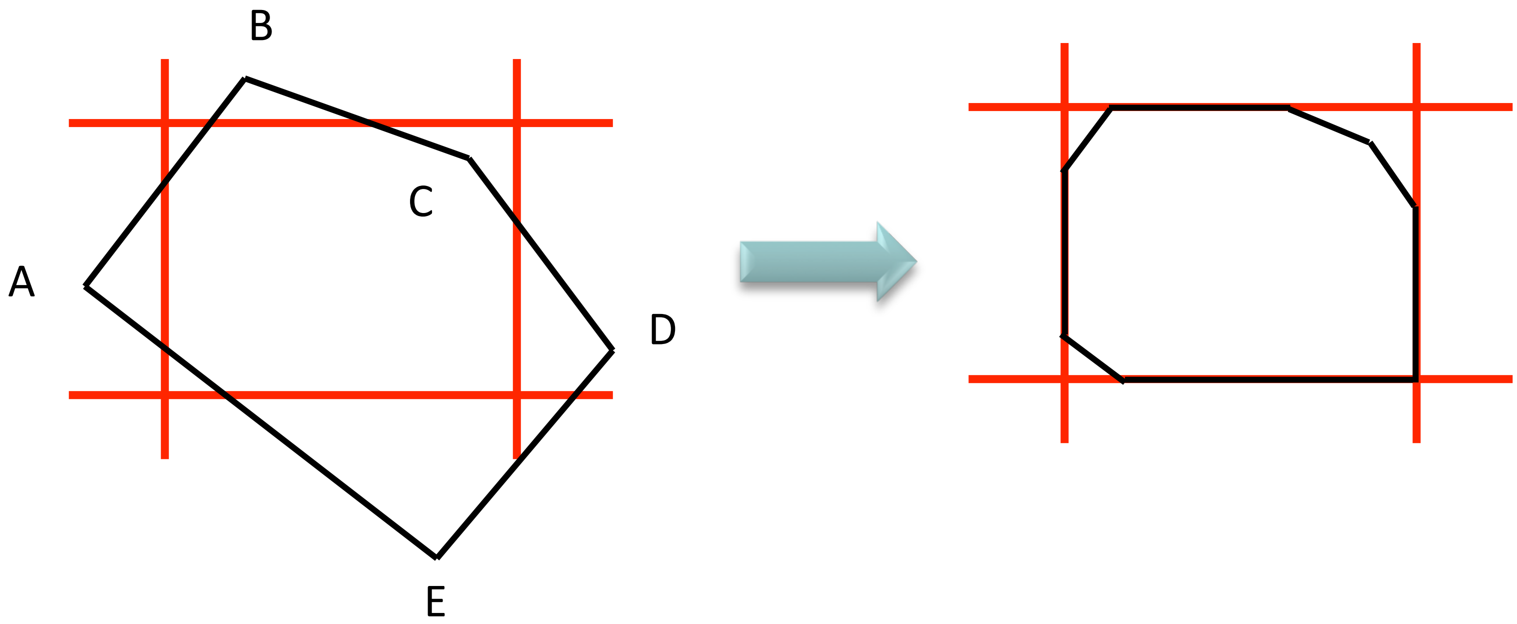
Princípio geral

- Processamento vértice a vértice do polígono
- Recorte de cada aresta PQ do polígono, mas garantindo um resultado fechado
- Considerando Q o vértice corrente, e P o anterior, adiciona-se P à lista de saída dos vértices se tal for apropriado e de acordo com 4 situações decorrentes do posicionamento da aresta PQ relativamente ao limite da janela
- O algoritmo aplica-se por fases sequenciais. Por exemplo, seguindo a ordem: Recorte à esquerda, recorte acima, recorte à direita, recorte abaixo

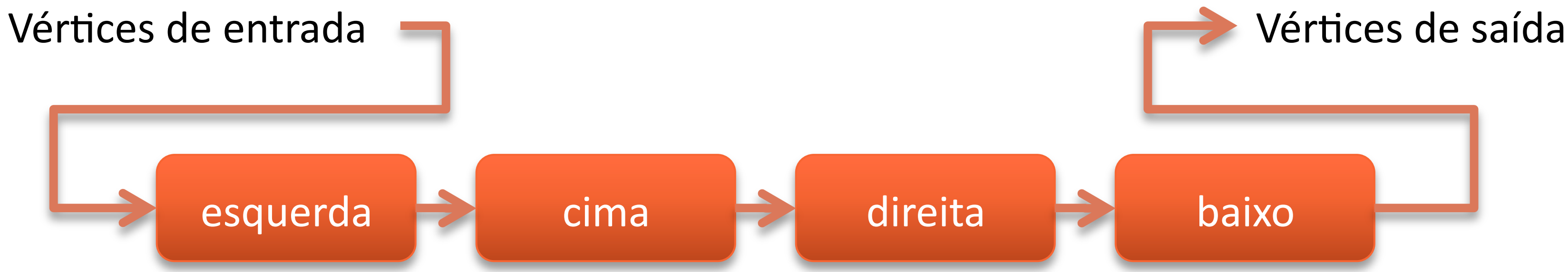
Entrada/saída do algoritmo

- Entrada: lista ordenada dos vértices do polígono
- Saída: lista de vértices do polígono recortado, podendo ser vértices iniciais ou novos

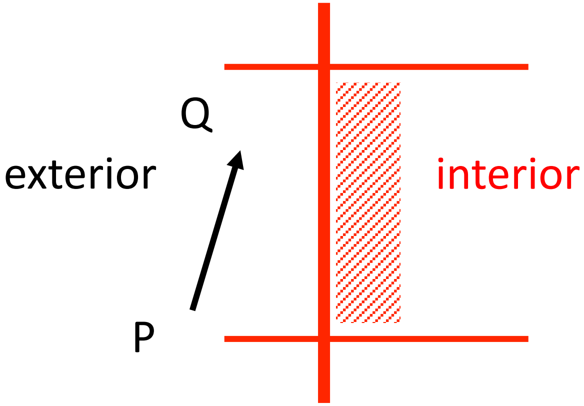
Recorte do polígono, com resultado fechado



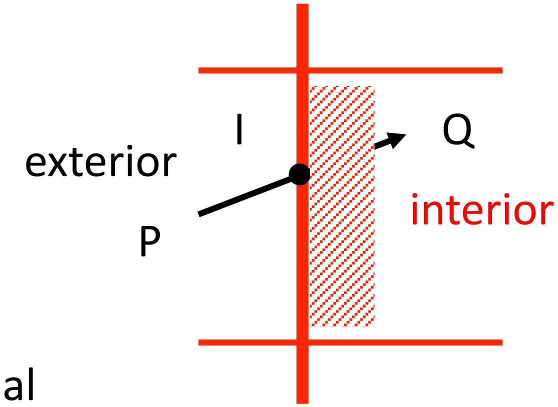
Fases de recorte sequenciais



4 situações decorrentes do posicionamento da aresta PQ relativamente ao limite da janela (ex: esquerda)

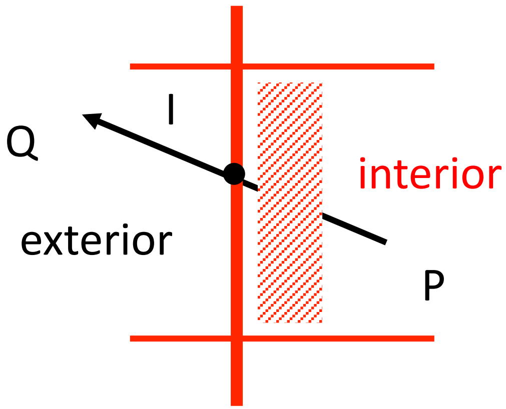


saída: { }

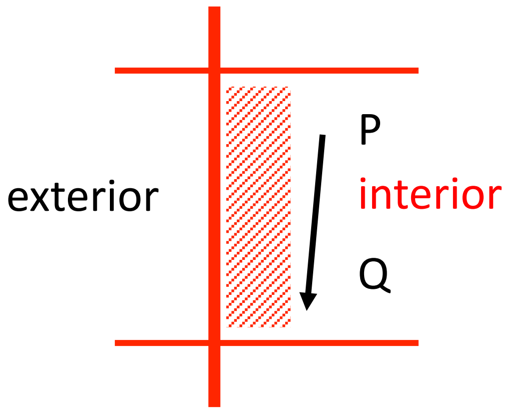


saída: { I, Q }

P é o vértice inicial

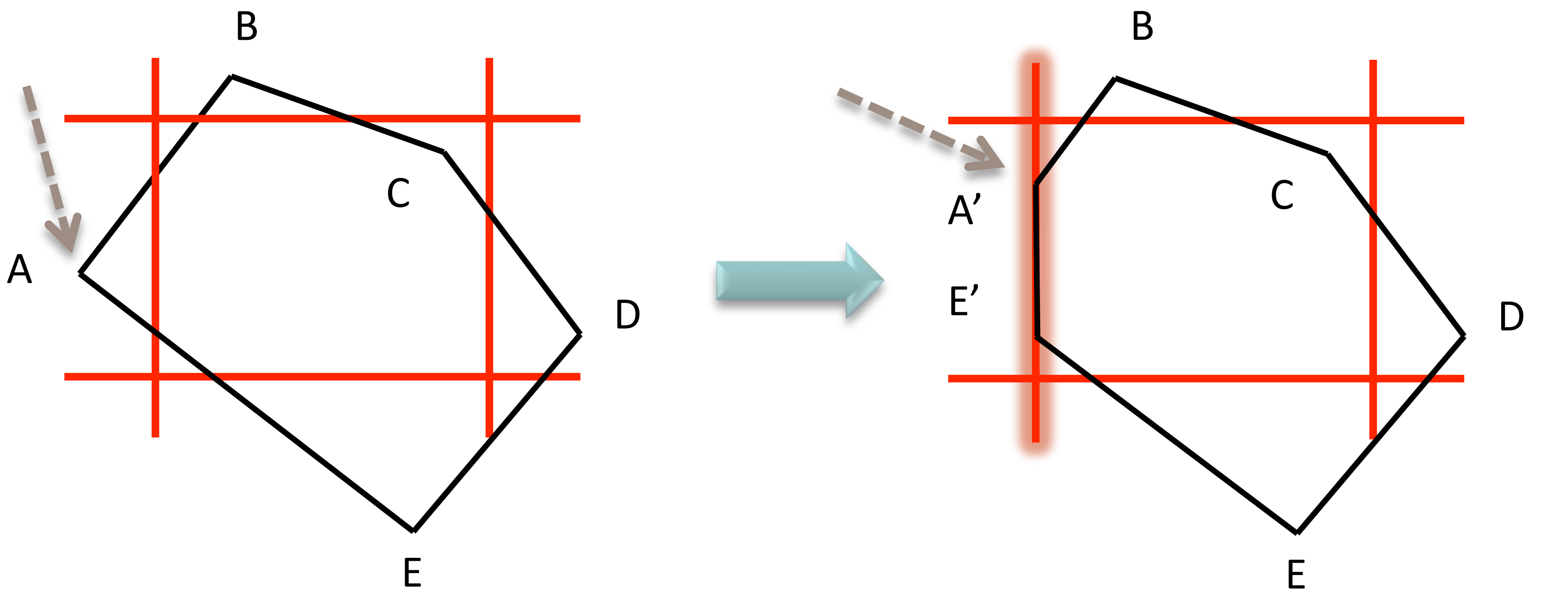


saída: { I }



saída: { Q }

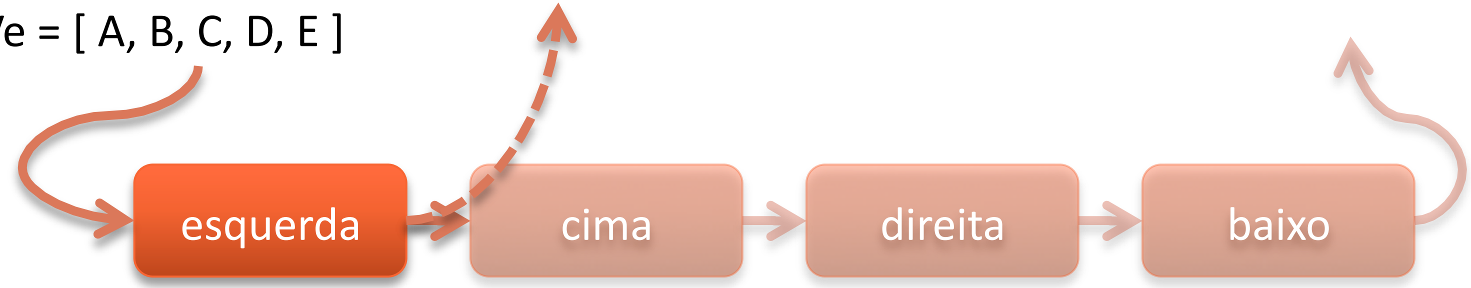
Exemplo

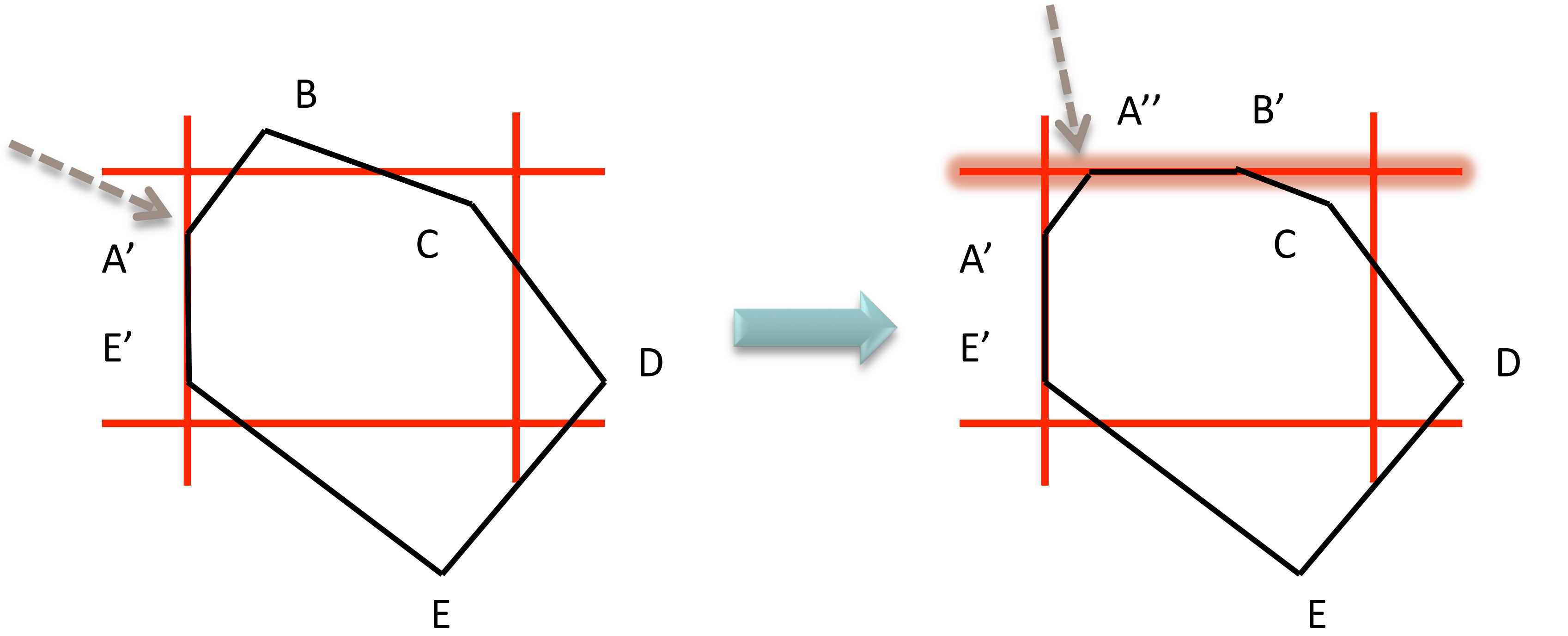


~~Vs_1 = [B, C, D, E, E’, A’]~~

Vs_1 = [A’, B, C, D, E, E’]

Ve = [A, B, C, D, E]

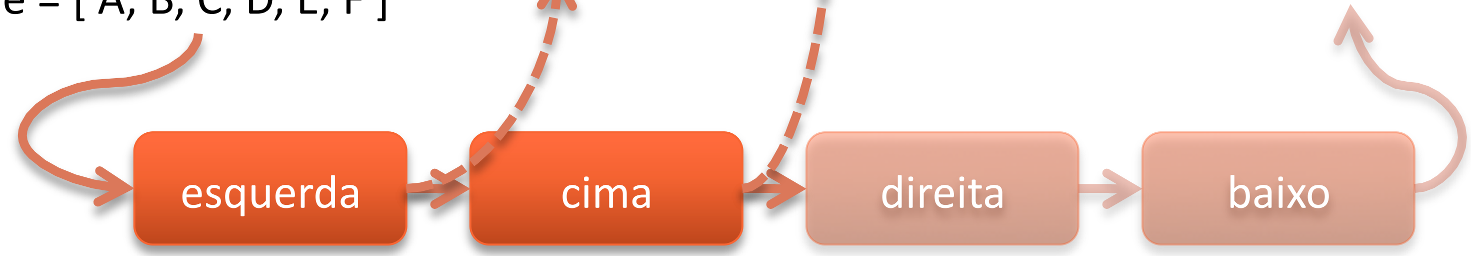


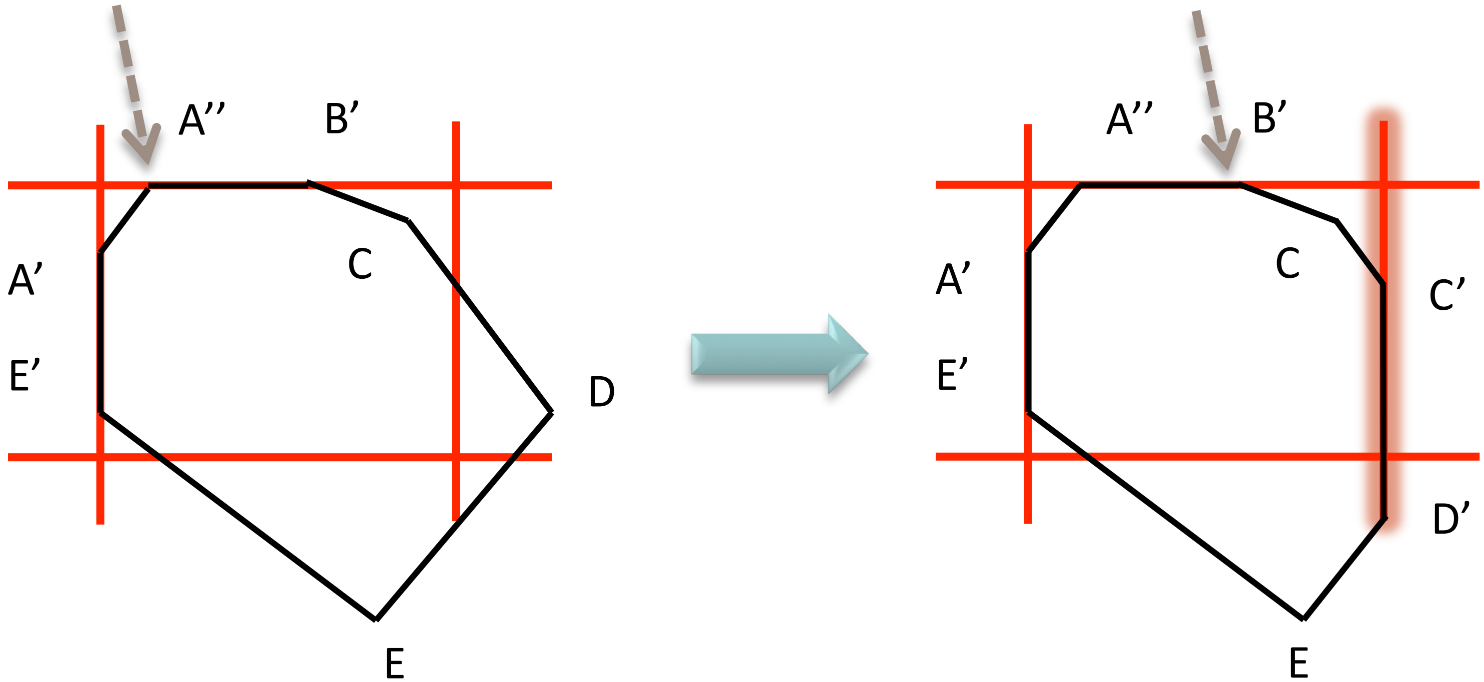


Vs_2 = [A'', B', C, D, E, E', A']

Vs_1 = [A', B, C, D, E, E']

Ve = [A, B, C, D, E, F]





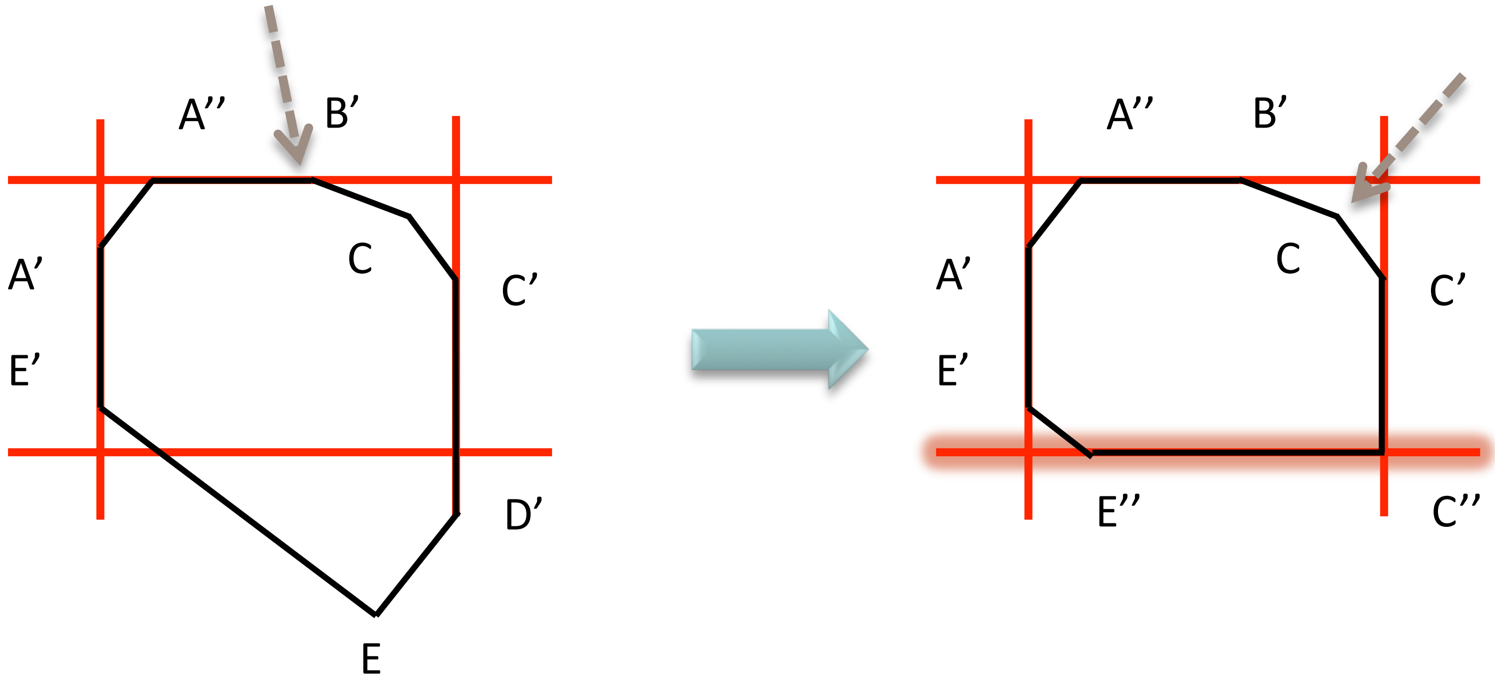
Vs_3 = [B', C, C', D', E, E', A', A'']

Vs_2 = [A'', B', C, D, E, E', A']

Vs_1 = [A', B, C, D, E, E']

Ve = [A, B, C, D, E, F]

esquerda → cima → direita → baixo



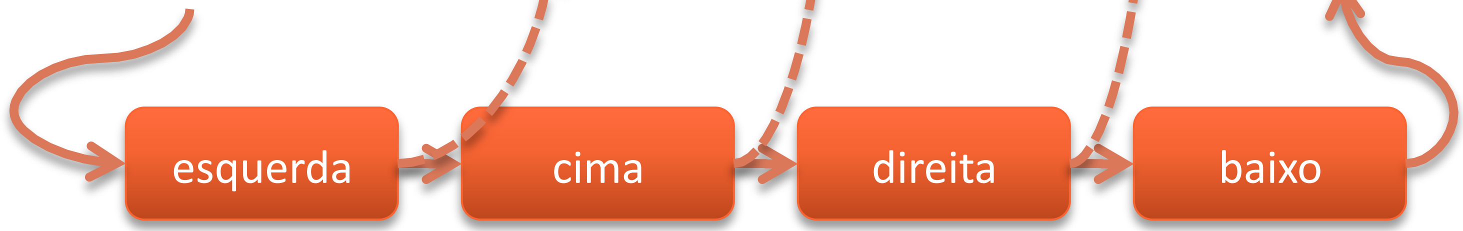
Vs_3 = [B', C, C', D', E, E', A', A'']

Vs_2 = [A'', B', C, D, E, E', A']

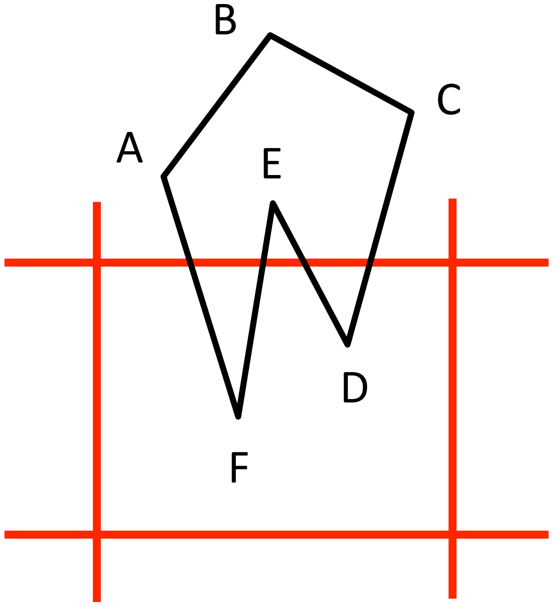
Vs_1 = [A', B, C, D, E, E']

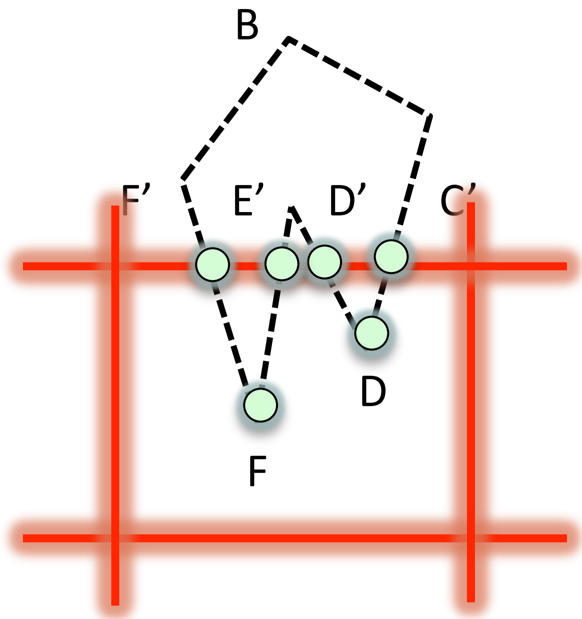
Ve = [A, B, C, D, E, F]

Vs = [C, C', C'', E'', E', A', A'', B']



No caso de polígonos não convexos, o algoritmo de Sutherland-Hodgman pode gerar arestas estranhas, que devem ser analisadas posteriormente (este problema não existe para os convexos)





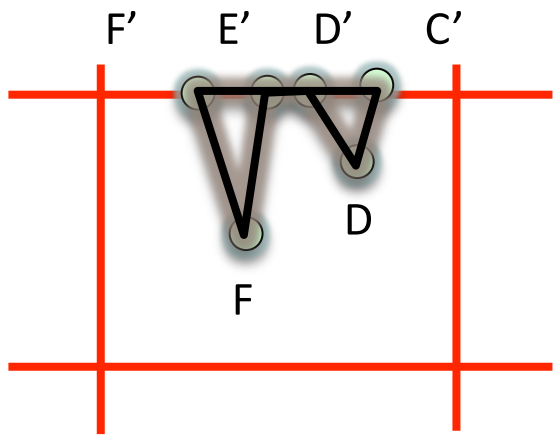
Ve = [A, B, C, D, E, F]

Vs_1 = [B, C, D, E, F, A]

Vs_2 = [C’, D, D’, E’, F, F’]

Vs_3 = [D, D’, E’, F, F’, C’]

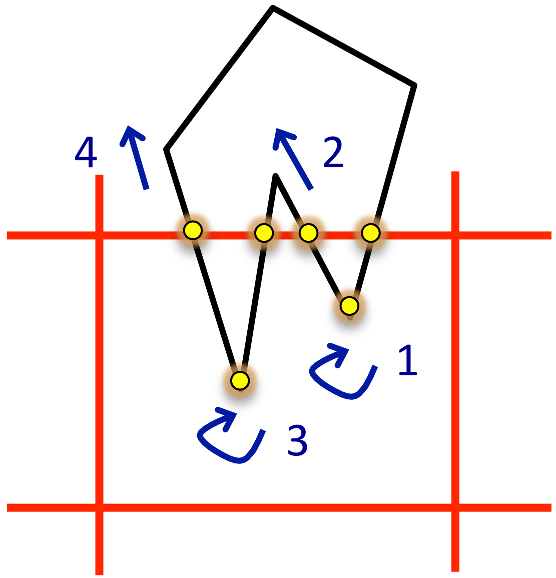
Vs = [D’, E’, F, F’, C’, D]

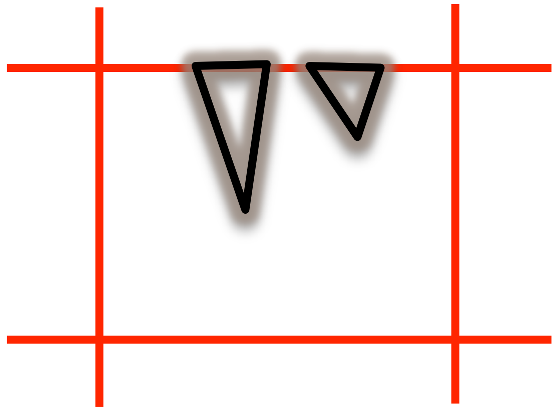


Neste exemplo, D’E’ e F’C’ são arestas estranhas

Remoção de arestas estranhas

- Seguindo o sentido indicado, se for encontrada uma intersecção com a aresta e se a aresta em percurso estiver a
  - entrar na janela, então memoriza-se a intersecção (inclui a aresta) e continua-se o percurso
  - sair da janela, então memoriza-se a intersecção (inclui a aresta) e o percurso é desviado para a direita, continuando pelas arestas da janela até à intersecção seguinte e retomando o polígono a recortar

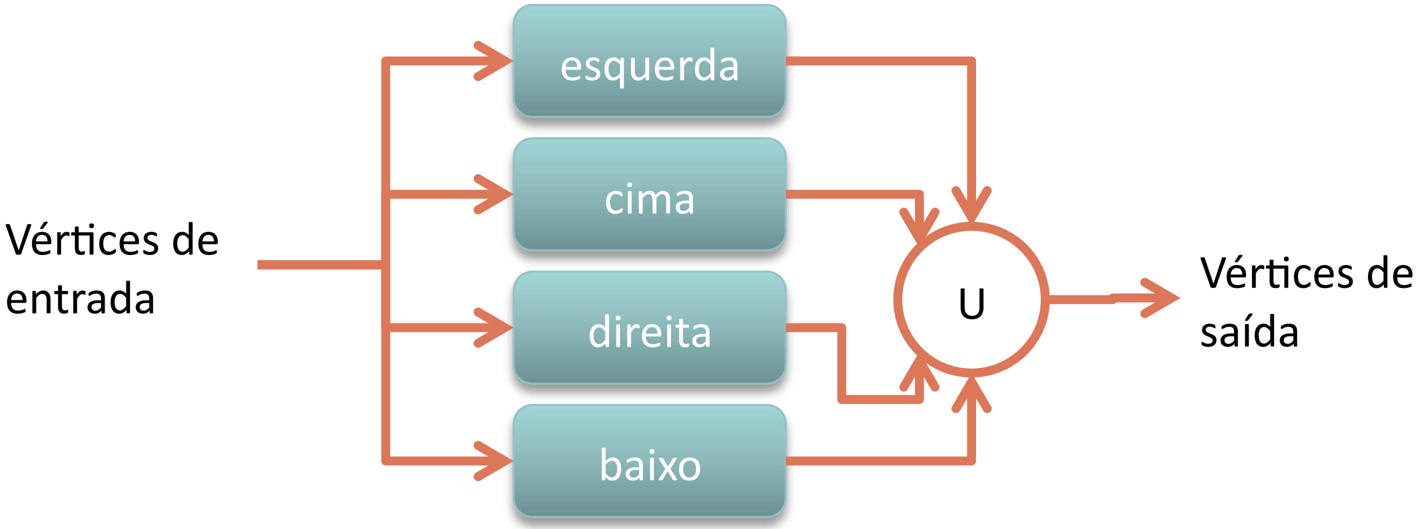




Programa de demonstração

Recorte exterior de polígonos (*shielding or blanking*)

Conjunção lógica dos vários resultados de recorte exterior em cada fronteira



Exemplo