

1/1/2016

Computação Gráfica

Recorte 2D

Dante Alves Medeiros Filho
UNIVERSIDADE ESTADUAL DE MARINGÁ

Universidade Estadual de Maringá

Departamento de Informática

Computação Gráfica

CG-07-RECORTE

NOTAS DE AULA

Prof. Dr. Dante Alves Medeiros Filho

2016

Sumário

7. Recorte	02
7.1 Recorte Bidimensional	02
7.1.1 Recorte de Pontos	04
7.1.2 Recorte de Segmentos de Retas	05
7.1.2.1 Método da Força Bruta	07
7.1.2.1.1 Cálculo das Intersecções com a Equação Explícita da Reta	07
7.1.2.1.2 Cálculo das Intersecções com a Equação Paramétrica da Reta	14
7.1.2.1.2.1 Generalização do Cálculo Paramétrico para Janelas Retangulares	22
7.1.2.1.3 Síntese sobre o Algoritmo da Força Bruta	23
7.1.2.2 Método da Cohen e Sutherland	23
7.1.2.2.1 Síntese Vantagens e Desvantagem do Algoritmo de Cohen e Sutherland	35
7.1.2.3 Algoritmo de Cyrus-Beck	36
7.1.2.3.1 Síntese sobre o Algoritmo de Cyrus Beck	50
7.1.2.4 Algoritmo de Liang e Barsky	50
7.1.2.4.1 Síntese sobre o Algoritmo de Liang e Barsky	58
7.1.2.5 Algoritmo da Divisão pelo Ponto Médio	58
7.1.2.5.1 Síntese sobre o Algoritmo do Ponto Médio	70
7.1.2.6 Algoritmo de Nicholl-Lee-Nicholl	71
7.1.3 Recorte de Polígonos	78
7.1.3.1 O método de Sutherland e Hodgman	79
7.1.3.1.1 Síntese sobre o Método de Sutherland e Hodgman	91
Bibliografia Recomendada	94

7. Recorte

Em muitas situações na computação gráfica é necessário restringir a visualização de objetos gráficos ou parte deles em um determinado dispositivo de saída gráfica. Ao processo de determinar o que será mostrado, e o que deverá ser omitido, dá-se o nome de **recorte**. Este procedimento pode ser custoso e o estudo de como realizá-lo visa a eficiência de sistemas computacionais gráficos. Pode ocorrer em vários momentos do pipeline de visualização em um contexto bidimensional ou tridimensional.

7.1 Recorte Bidimensional

No caso bidimensional o recorte pode ser feito com relação a vários objetos gráficos. Neste texto, vamos trabalhar o recorte de pontos, linhas e polígonos. Esta operação geralmente é feita contra uma região retangular (Figura 1).

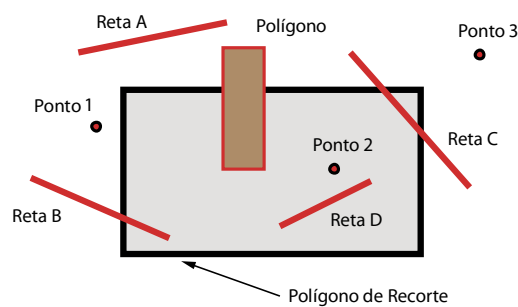


Figura 1: Objetos gráficos bidimensionais e uma **região de recorte** retangular

O problema reside em determinar os objetos gráficos ou parte deles que se encontram no interior da região de recorte para que possam ser desenhados (Figura 2). Esta operação é conhecida como **clipping**.

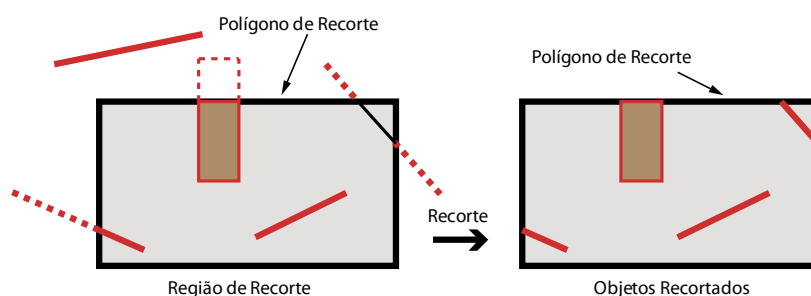


Figura 2: Recorte de objetos gráficos bidimensionais contra uma **região de recorte** retangular

Os objetos gráficos, bem como o polígono de recorte devem ser formalizados sob a égide de um determinado sistema de coordenadas. Neste sentido, o procedimento de recorte pode ser feito utilizando um sistema de coordenadas do mundo (**WCS – World Coordinate System**) ou um sistema de coordenadas de dispositivos (**DCS - Device Coordinate System**). Temos assim, dois caminhos para a realização desta operação. É importante salientar que o termo **Janela** é utilizado para uma região de recorte definida em **WCS** e **Viewport** em **DCS**.

Primeiro Caminho:

Os objetos gráficos e o polígono de recorte são definidos em **WCS**. O procedimento de recorte é feito neste sistema (Figura 3). Depois de realizado o recorte é feita uma transformação **janela-viewport**.

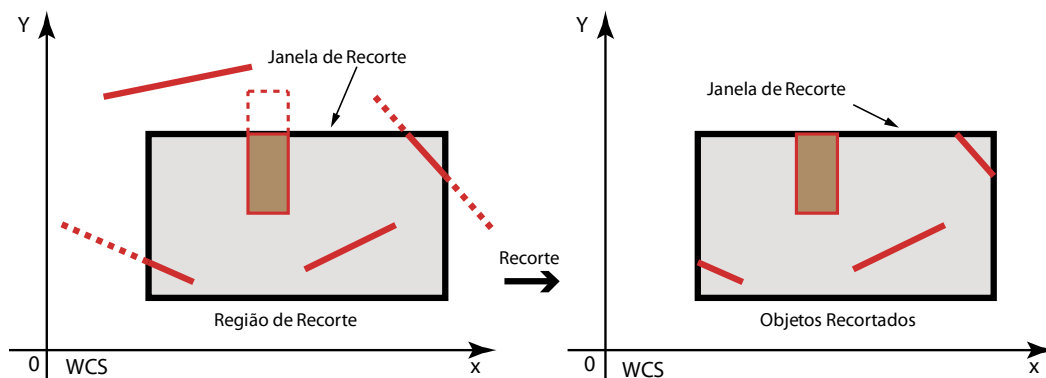


Figura 3: Recorte realizado em World Coordinate System (WCS).

Após a realização do recorte em **WCS**, é feita uma transformação **janela-viewport** que leva o resultado para o sistema de coordenadas do dispositivo gráfico (**DCS – Device Coordinate System**) ou para um sistema normalizado (**NDCS – Normalized Device Coordinate System**), (Figura 4).

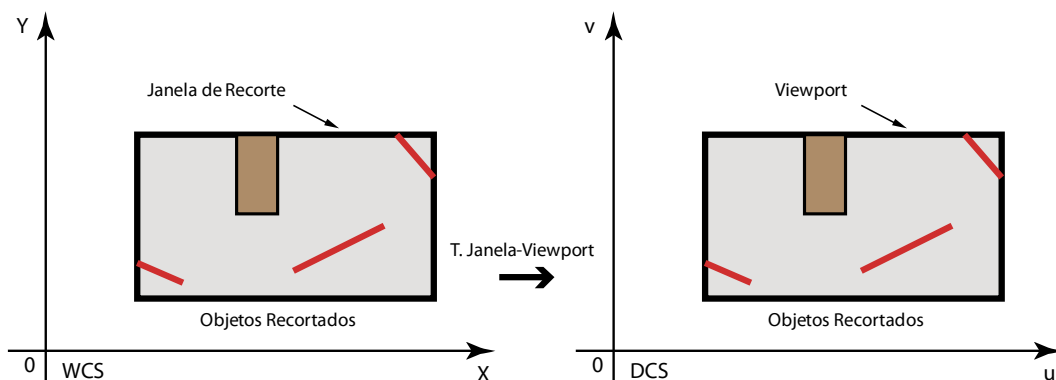


Figura 4: Transformação **Janela-Viewport** (WCS para NDCS ou DCS)

Segundo Caminho:

Os objetos gráficos e o polígono de recorte são definidos em **WCS**. Inicialmente faz-se a transformação **Janela-viewport** sem executar o recorte (figura 5).

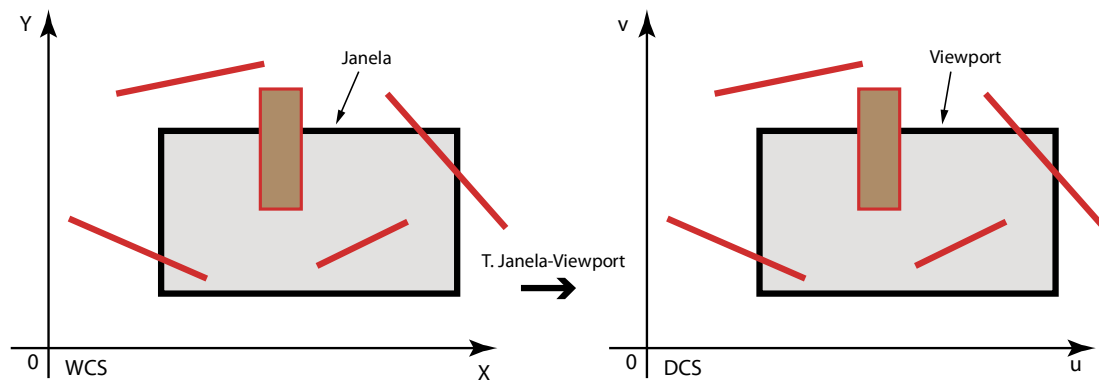


Figura 5: Transformação **Janela-Viewport** (**WCS** para **DCS** ou **NDC**)

Em seguida realizamos o procedimento de recorte. Note que agora ele é feito no sistema de coordenadas do dispositivo, **DCS** (Figura 6).

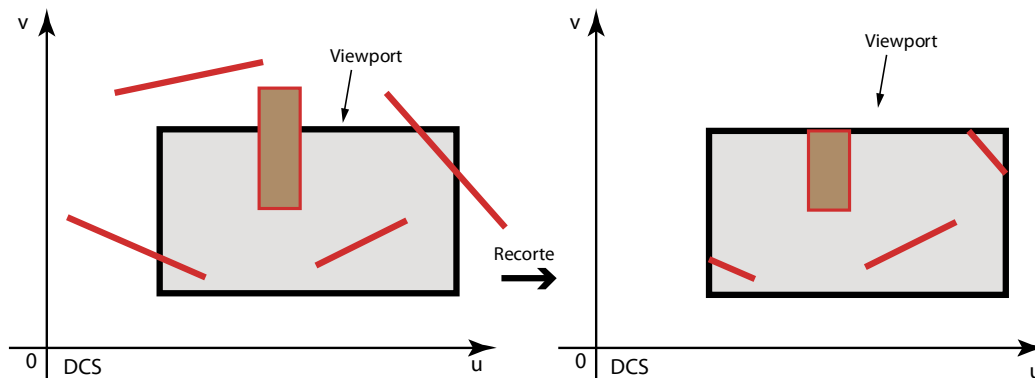


Figura 6: Recorte em Coordenadas do Dispositivo (**DCS**) ou em um Sistema Normalizado (**NDCS**)

Vamos iniciar este estudo determinando pontos que estejam dentro ou fora de um polígono de recorte, para depois, apresentar como isto pode ser feito com seguimentos de retas e polígonos.

7.1.1 Recorte de Pontos

Considere um ponto $P(x,y)$ especificado em um determinado sistema de coordenadas $\{0,x,y\}$ e uma região de recorte retangular conforme mostra a Figura 7.

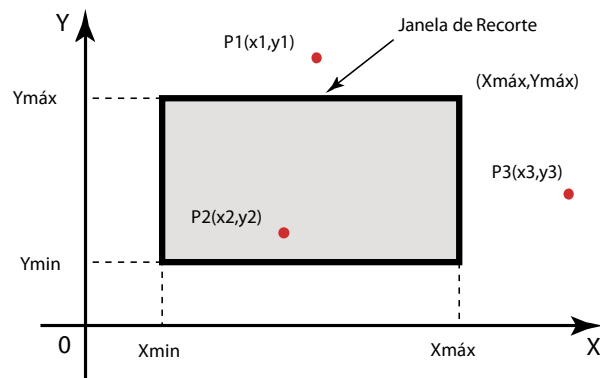


Figura 7: pontos e uma janela de recorte definidos em um sistema de coordenadas $\{0,x,y\}$

Um ponto com coordenadas (X,Y) encontra-se dentro de um retângulo de recorte quando se verificarem, simultaneamente, duas condições:

$$x_{min} \leq x \leq x_{max}$$

$$y_{min} \leq y \leq y_{max}$$

Os pontos que não satisfizerem estas condições estão fora da área de visualização, portando com a operação de recorte não serão visualizados (figura 8).

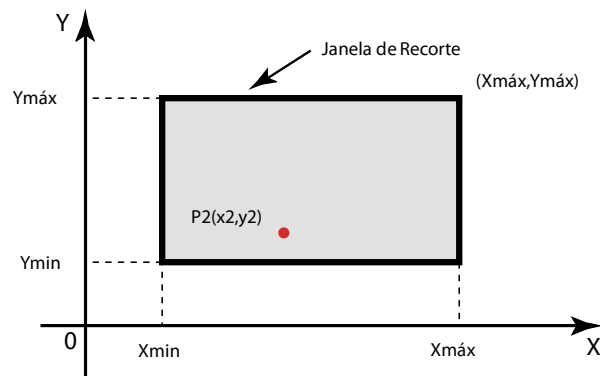


Figura 8: recorte dos pontos P1 e P3

7.1.2 Recorte de Segmentos de Retas

O recorte de segmentos de retas é muito utilizado em computação gráfica e geralmente é feito contra uma janela retangular. Observe a figura 9 e note que alguns segmentos de reta ultrapassam os limites da **janela de recorte** retangular especificada. Alguns estão

totalmente dentro do retângulo de visualização (**janela de recorte**), outras, parcialmente e algumas, totalmente fora dele.

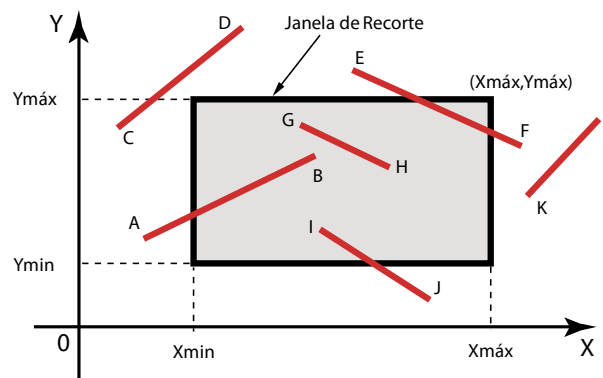


Figura 9: recorte de segmentos de retas

A operação de recorte consiste em determinar quais segmentos de reta ou parte deles serão visíveis considerando o retângulo de recorte. É importante notar que alguns segmentos estão totalmente fora da região de visualização (**janela de recorte**). O seguimento definido pelos pontos GH está totalmente imerso na **janela de recorte**. Alguns segmentos estão parcialmente dentro e fora da **janela de recorte** (Figura 9). Após o recorte, teremos como resultado, o ilustrado na figura 10.

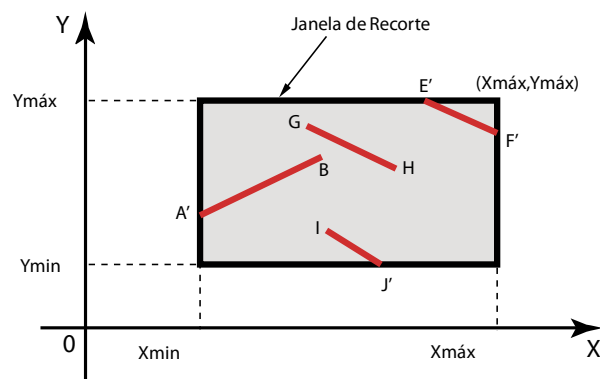


Figura 10: segmentos de retas recortadas em função da **janela de recorte**

Para fazer o recorte é preciso verificar quais segmentos de reta ou parte deles serão mostrados na **janela de recorte**. Para isto é preciso calcular a intersecção desses segmentos com os segmentos que constituem a **janela de recorte**. Este procedimento pode exigir grande esforço computacional. Vários trabalhos foram realizados na tentativa de tornar este procedimento mais eficiente. Inicialmente vamos mostrar como o recorte de segmentos de reta pode ser feito calculando as intersecções de cada segmento com as arestas da **janela**

de recorte e depois, verificar se estes pontos, estão dentro ou fora deste polígono. Em seguida, apresentaremos algoritmos de recorte mais elaborados que buscam mais eficiência neste processo, bem como, algoritmos que possuem contribuições históricas e didáticas para a sua compreensão.

7.1.2.1 Método da Força Bruta

Este método consiste em determinar a intersecção dos segmentos de reta com todas as arestas da **janela de recorte**. As intersecções podem ser calculadas por meio da função $y = mx + b$ ou da equação paramétrica da reta. Devido a seu acentuado esforço computacional é conhecido como **algoritmos da força bruta, direto** ou **exaustivo**.

7.1.2.1.1 Cálculo das Intersecções com a Equação Explícita da Reta

De uma forma pragmática vamos ilustrar o cálculo de intersecções de um segmento de reta definido por dois pontos distintos no espaço por meio de um exemplo.

Exemplo 01: Considere a figura 11 onde temos uma **janela de recorte** retangular definida pelos extremos $X_{\min}=2$, $Y_{\min}=1$ e $X_{\max}=7$, $Y_{\max}=4$ e um segmento de reta definido pelo pontos $A(1,1.5)$, $B(4,3)$.

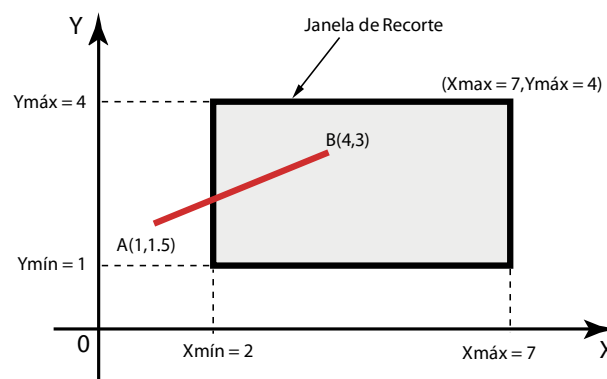


Figura 11: Linha AB a ser recortada pela **Janela de Recorte**

Vamos calcular a intersecção do segmento de reta AB com todas as arestas da **janela de recorte**. De forma genérica o segmento de reta pode ser definido pela seguinte função:

$$y = mx + b$$

Aplicada aos pontos $A(x_A, y_A)$ e $B(x_B, y_B)$, temos:

$$y_{AB} = m_{AB}x + b$$

$$A(x_A, y_A) = (1, 1.5)$$

$$B(x_B, y_B) = (4, 3)$$

$$m_{AB} = \frac{y_B - y_A}{x_B - x_A} = \frac{1}{2}$$

$$y_{AB} = 0.5x + b$$

Da **janela de recorte** temos:

$$x_{min} = 2.0; \quad y_{min} = 1.5$$

$$x_{max} = 7.0; \quad y_{max} = 4.0$$

Iniciaremos os cálculos com a determinação da intersecção do segmento de reta **AB** com a **aresta superior** da **Janela de Recorte**.

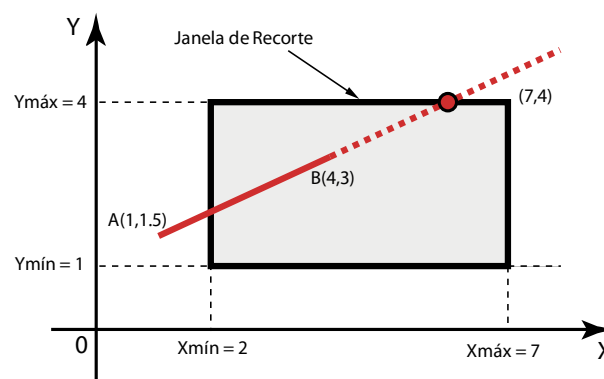


Figura 12: Intersecção do segmento de reta AB com a **aresta superior** da **Janela de Recorte**

O ponto de intersecção do segmento de reta AB com a **aresta superior** da **janela de recorte** (I_{as}), possui as seguintes coordenadas:

$$I_{as} = (x_{as}, y_{m\acute{a}x})$$

$$y_{m\acute{a}x} = 4$$

Aplicando a função $y = mx + b$ no ponto A:

$$y_A = m_{AB} \cdot x_A + b$$

$$b = y_A - m_{AB} \cdot x_A$$

Sabemos que $y_{as} = y_{m\acute{a}x}$, aplicado a equação da reta:

$$y_{as} = y_{m\acute{a}x} = m_{AB} \cdot x_{as} + b$$

Substituindo b:

$$y_{m\acute{a}x} = m_{AB} \cdot x_{as} + (y_A - m_{AB} \cdot x_A)$$

$$y_{m\acute{a}x} = m_{AB} \cdot x_{as} + y_A - m_{AB} \cdot x_A$$

$$y_{m\acute{a}x} = m_{AB} \cdot (x_{as} - x_A) + y_A$$

$$x_{as} = x_A + \frac{1}{m_{AB}}(y_{m\acute{a}x} - y_A); \text{ com } m_{AB} \neq 0$$

$$x_{as} = 1 + 2 \cdot (4 - 1.5) = 6.0$$

O ponto de intersecção com a aresta superior é o ponto $I_{as} = (6,4)$, ilustrado na figura 13:

$$I_{as} = (x_{as}, y_{m\acute{a}x}) = (6,4)$$

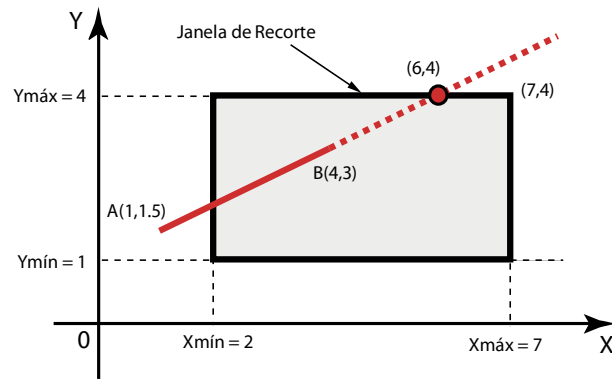


Figura 13: (6,4) é o ponto de intersecção do segmento AB com a **aresta superior** da **Janela de Recorte**

Vamos agora determinar a intersecção do segmento de reta AB com a **aresta inferior** da **Janela de Recorte** (figura 14).

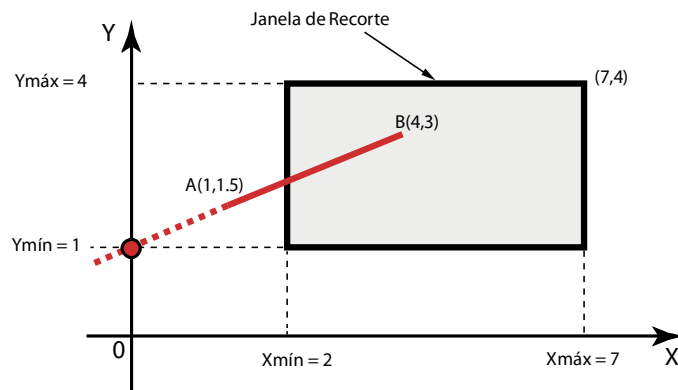


Figura 14: Intersecção do segmento de reta AB com a **aresta inferior** da **Janela de Recorte**

O ponto de intersecção do segmento de reta AB com a **aresta inferior** da **janela de recorte** (I_{ai}) terá as seguintes coordenadas:

$$I_{ai} = (x_{ai}, y_{mín})$$

$$y_{mín} = 1$$

Analogamente ao cálculo realizado para encontrar o valor da abscissa do ponto de intersecção da aresta superior, temos para a aresta inferior:

$$x_{ai} = x_A + \frac{1}{m_{AB}}(y_{mín} - y_A); \text{ com } m_{AB} \neq \infty$$

$$x_{ai} = 1 + 2 \cdot (1 - 1.5) = 0$$

$$I_{ai} = (x_{ai}, y_{\min}) = (0, 1)$$

O ponto de intersecção com a **aresta inferior** da **janela de recorte** é o ponto $(x_{ai}, y_{\min}) = (0, 1)$ ilustrado na figura 15.

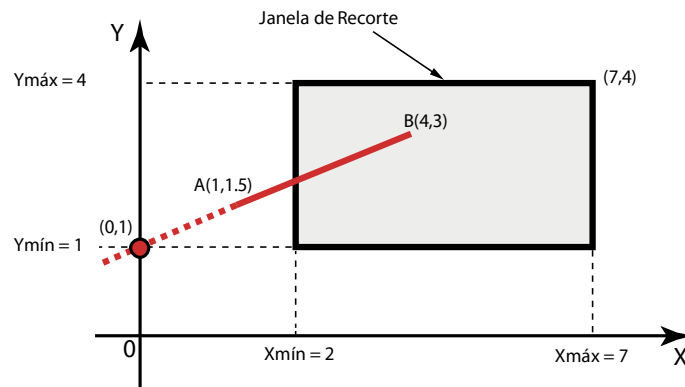


Figura 15: (0,1) é o ponto de intersecção do segmento AB com a **aresta inferior** da **Janela de Recorte**

Vamos agora determinar a intersecção do segmento de reta AB com a **aresta direita** da **Janela de Recorte** (figura 16).

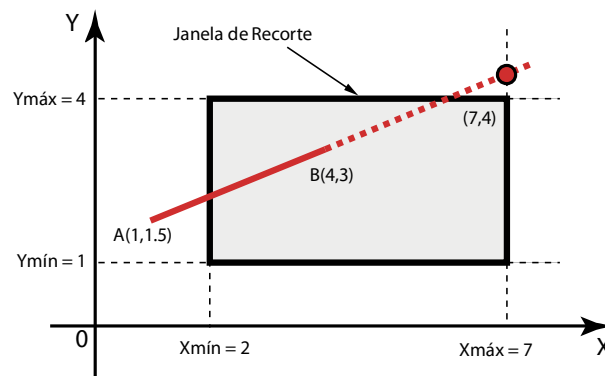


Figura 16: Intersecção do segmento de reta AB com a **aresta direita** da **Janela de Recorte**

Temos que encontrar um ponto de intersecção da reta AB com a **aresta direita** ($I_{aresta\ direita} = I_{ad}$) da **janela de recorte** que tenha coordenadas:

$$I_{ad} = (x_{máx}, y_{ad})$$

$$x_{máx} = 7$$

$$y_{ad} = m_{AB}(x_{m\acute{a}x} - x_A) + y_A; \quad \text{com } m_{AB} \neq \infty$$

$$y_{ad} = 0.5(7 - 1) + 1.5 = 4.5$$

$$I_{ad} = (x_{m\acute{a}x}, y_{ad}) = (7, 4.5)$$

O ponto de intersecção com a **aresta direita** da **janela de recorte** é o ponto $(x_{m\acute{a}x}, y_{ad}) = (7, 4.5)$, ilustrado na figura 17.

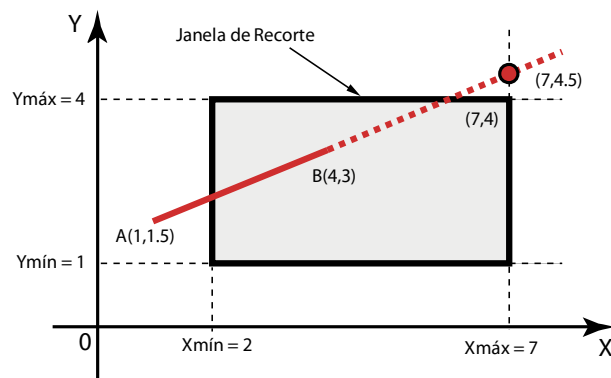


Figura 17: (7,4.5) é o ponto de intersecção do segmento AB com a **aresta direita** da **Janela de Recorte**

Cálculo do ponto de intersecção do segmento de reta AB com a **aresta esquerda** da **Janela de Recorte** (I_{ae}).

$$I_{ae} = (x_{m\acute{i}n}, y_{ae})$$

$$x_{min} = 2,0$$

$$y_{ae} = m_{AB} \cdot x_{m\acute{i}n} + b$$

Escolhendo o ponto A temos:

$$y_A = m_{AB} \cdot x_A + b$$

$$b = y_A - m_{AB} \cdot x_A$$

$$y_{ae} = m_{AB} \cdot x_{m\acute{i}n} + (y_A - m_{AB} \cdot x_A)$$

$$y_{ae} = m_{AB} \cdot x_{m\acute{i}n} + y_A - m_{AB} \cdot x_A$$

$$y_{ae} = m_{AB}(x_{\min} - x_A) + y_A \quad \text{para } m \neq \infty$$

$$y_{ae} = 0.5(2 - 1) + 1.5$$

$$y_{ae} = 2,0$$

O ponto de intersecção com a aresta esquerda da **janela de recorte** é o ponto $I_{ae} = (x_{\min}, y_{ae}) = (2,2)$, ilustrado na figura 18.

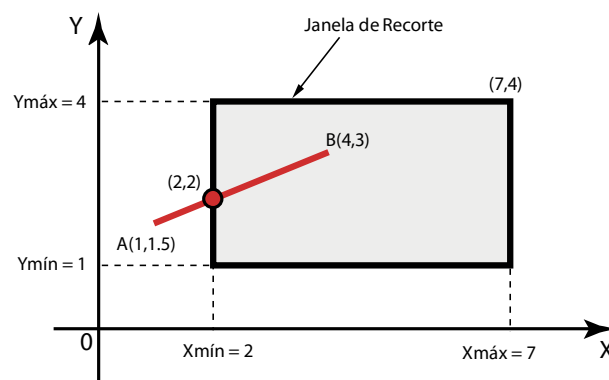


Figura 18: Intersecção do segmento de reta AB com a aresta esquerda da **Janela de Recorte**

Temos agora que ver se os pontos calculados estão no interior da **janela de recorte** ou não. Só serão mostrados os segmentos dentro dos limites da **janela de recorte** (Figura19). Ou seja, para:

$$x_{\min} \leq x \leq x_{\max}$$

$$y_{\min} \leq y \leq y_{\max}$$

Analisando os pontos de intersecção do segmento de reta AB com as arestas da **janela de recorte** podemos notar que apenas os pontos $(x_{\min}, y_{ae}) = (2,2)$ e $(x_B, y_B) = (4,3)$ satisfazem as condições acima.

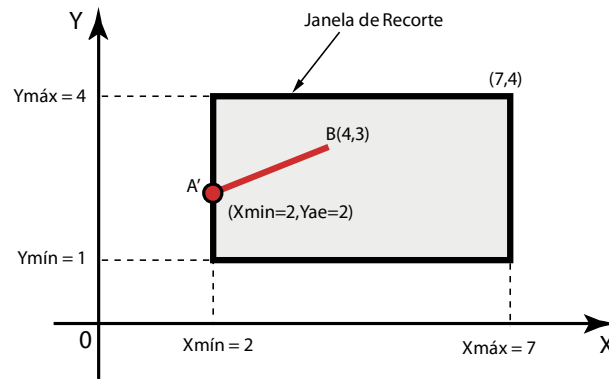


Figura 19: Segmento de reta AB recortado

7.1.2.1.2 Cálculo das Intersecções com a Equação Paramétrica da Reta

Os cálculos da intersecção do segmento de reta AB com as arestas da **janela de recorte**, podem ser realizados com a equação paramétrica da reta. Considere uma segmento de reta como o da figura 20.

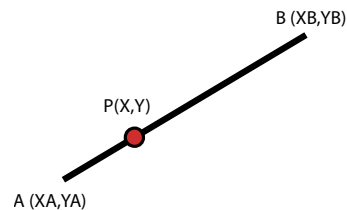


Figura 20: Segmento de reta AB com um ponto genérico (X,Y)

A equação paramétrica em 2D do segmento de reta AB pode, genericamente, ser escrita da seguinte forma:

$$P(t_{AB}) = P_A + (P_B - P_A) \cdot t_{AB}$$

$$x = x_A + at_{AB}$$

$$y = y_A + bt_{AB}$$

Sendo :

$$a = (x_B - x_A)$$

$$b = (y_B - y_A)$$

$$t_{AB} = [0,1]$$

Temos então:

$$x = x_A + (x_B - x_A)t_{AB}; \text{ para } t_{AB} \ 0 \leq t \leq 1$$

$$y = y_A + (y_B - y_A)t_{AB}; \text{ para } t_{AB} \ 0 \leq t \leq 1$$

Em função de t podemos descrever o segmento de reta AB, (figura 21)

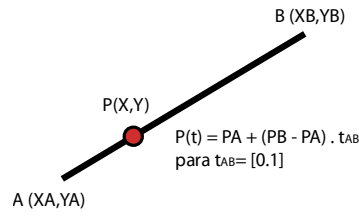


Figura 21: Segmento de reta AB em função do parâmetro t_{AB}

$$P(t_{AB}) = P_A + (P_B - P_A) \cdot t_{AB} ; \quad 0 \leq t_{AB} \leq 1$$

$$t_{AB} = \frac{P(t_{AB}) - P_A}{P_B - P_A}$$

Vamos também descrever as arestas da **janela de recorte** como segmentos de reta por meio de suas equações paramétricas.

Aresta superior (as):

$$\begin{aligned} x_{as} &= x_{min} + (x_{máx} - x_{min})t_{as} \\ y_{as} &= y_{máx} + (y_{máx} - y_{máx})t_{as} = y_{máx} \end{aligned}$$

Aresta Inferior (ai):

$$\begin{aligned} x_{ai} &= x_{min} + (x_{máx} - x_{min})t_{ai} \\ y_{ai} &= y_{mín} + (y_{mín} - y_{mín})t_{ai} = y_{mín} \end{aligned}$$

Aresta direita (ad):

$$\begin{aligned} x_{ad} &= x_{máx} + (x_{máx} - x_{máx})t_{ad} = x_{máx} \\ y_{ad} &= y_{mín} + (y_{máx} - y_{mín})t_{ad} \end{aligned}$$

Aresta esquerda (ae):

$$\begin{aligned} x_{ae} &= x_{min} + (x_{min} - x_{min})t_{ae} = x_{min} \\ y_{ae} &= y_{mín} + (y_{máx} - y_{mín})t_{ae} \end{aligned}$$

Observe que $t_{as}, t_{ai}, t_{ad}, t_{ae}$ variam entre 0 e 1.

Para ilustrar como pode ser feito o cálculo das intersecções de um segmento de reta AB com as arestas de uma **janela de recorte** com o uso de equações paramétricas vamos determinar as intersecções do exemplo anterior, só que agora, com o uso de tipo de equação.

Exemplo 02: Considere a figura 22. Neste contexto, vamos calcular as intersecções do segmento de reta AB com as arestas da janela retangular de recorte.

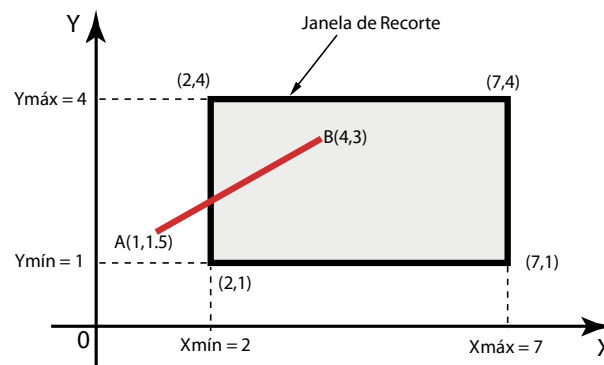


Figura 22: Segmento de reta AB a ser recortado

O retângulo de recorte é definido pelos seguintes vértices:

$$x_{mín} = 2.0$$

$$y_{mín} = 1.0$$

$$x_{máx} = 7.0$$

$$y_{máx} = 4.0$$

O segmento de reta AB é definido pelos pontos

$$x_A = 1.0$$

$$y_A = 1.5$$

$$x_B = 4.0$$

$$y_B = 3.0$$

Que geram as seguintes equações paramétricas:

$$x_{AB} = x_A + (x_B - x_A)t_{AB}$$

$$y_{AB} = y_A + (y_B - y_A)t_{AB}$$

Para os pontos A(1,1.5) e B(4,3):

$$x_{AB} = 1 + 3t_{AB}$$

$$y_{AB} = 1.5 + 1.5t_{AB}$$

Cálculo da intersecção do segmento AB com a aresta superior da janela de recorte (I_{as}).

As equações paramétricas da aresta superior da **janela de recorte** são:

$$x_{as} = x_{min} + (x_{máx} - x_{min})t_{as}$$

$$y_{as} = y_{máx}$$

Substituindo os valores:

$$x_{as} = 2 + 5t_{as}$$

$$y_{as} = 4$$

Como o ponto é de intersecção entre o segmento de reta AB e a **aresta superior da janela de recorte**, os valores de x e y destes pontos são iguais. Temos que:

$$x_{AB} = x_{as}$$

$$y_{AB} = y_{as}$$

$$1 + 3t_{AB} = 2 + 5t_{as}$$

$$1.5 + 1.5t_{AB} = 4$$

onde:

$$t_{AB} = 5/3$$

e t_{as} :

$$1 + 3\left(\frac{5}{3}\right) = 2 + 5t_{as}$$

$$6 = 2 + 5t_{as}$$

$$t_{as} = 0.8$$

Estes valores aplicados as equações paramétricas descritas, permitem o cálculo do ponto de intersecção entre o segmento AB e a aresta superior da **janela de recorte** denominada I_{as} , (Figura 23):

$$I_{as} = (x_{as}, y_{máx}) = (6,4)$$

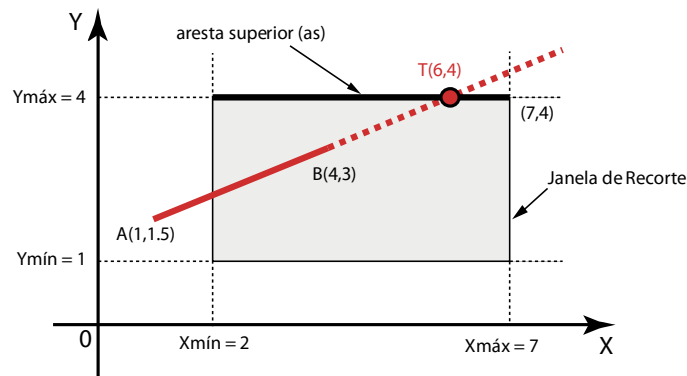


Figura 23: Intersecção de AB com a **aresta superior** (I_{as}) da **janela de recorte**

O valor de $t_{as} = 0.8$, indica que o ponto de intersecção está dentro da aresta superior, pois está entre 0 e 1. No entanto, o valor de $5/3$ para t_{AB} indica que o ponto de intersecção não se encontra no interior do seguimento de reta AB, pois está fora do intervalo $[0,1]$.

Cálculo da intersecção do segmento AB com a aresta inferior da janela de recorte (I_{ai})

Da aresta inferior da **janela de recorte** temos:

$$x_{ai} = x_{min} + (x_{máx} - x_{min})t_{ai}$$

$$y_{ai} = y_{mín}$$

Substituindo os valores:

$$x_{ai} = 2 + 5t_{as}$$

$$y_{ai} = 1$$

Como o ponto é de intersecção entre o segmento de reta AB e a aresta superior da **janela de recorte**, os valores de x e y destes pontos são iguais. Assim temos:

$$x_{AB} = x_{ai}$$

$$y_{AB} = y_{ai}$$

$$1 + 3t_{AB} = 2 + 5t_{ai}$$

$$1.5 + 1.5t_{AB} = 1$$

onde:

$$t_{AB} = -1/3$$

para t_{ai} :

$$1 + 3\left(-\frac{1}{3}\right) = 2 + 5t_{ai}$$

$$0 = 2 + 5t_{ai}$$

$$t_{ai} = -0.4$$

Substituindo os valores de t_{ai} e t_{AB} nas equações paramétricas podemos calcular o ponto de intersecção procurado (Figura 24):

$$I_{ai} = (x_{ai}, y_{mín}) = (0,1)$$

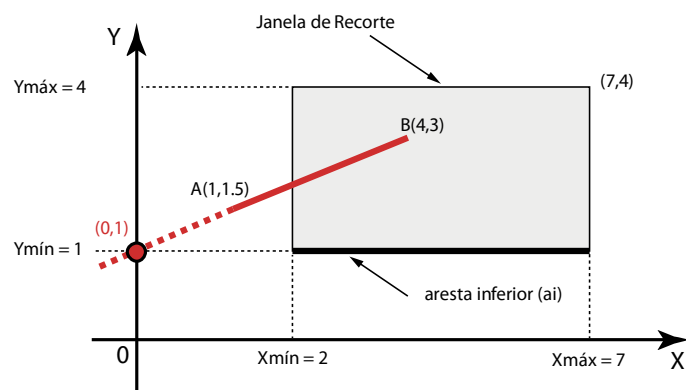


Figura 24: Intersecção de AB com a aresta inferior (ai) da janela de recorte

O valor de $t_{as} = -0.4$, indica que o ponto de intersecção não se encontra na aresta inferior do retângulo de recorte, pois não está entre 0 e 1. O valor de $-1/3$ para t_{AB} indica que o ponto de intersecção não se encontra no interior do seguimento de reta AB, pois t está fora do intervalo $[0,1]$.

Cálculo da intersecção do segmento AB com a aresta direita da janela de recorte (I_{ad})

Da aresta inferior da **janela de recorte** temos:

$$x_{ad} = x_{máx}$$
$$y_{ad} = y_{mín} + (y_{máx} - y_{mín})t_{ad}$$

Substituindo os valores:

$$x_{ad} = 7 + 0 \cdot t_{ad} = 7$$
$$y_d = 1 + 3t_{ad}$$

Como o ponto é de intersecção entre o segmento de reta AB e a **aresta direita da janela de recorte**, os valores de x e y destes pontos são iguais. Assim temos:

$$x_{AB} = x_{ad}$$
$$y_{AB} = y_{ad}$$

$$1 + 3t_{AB} = 7$$
$$1.5 + 1.5t_{AB} = 1 + 3t_{ad}$$

e:

$$t_{AB} = 2$$

Para t_{ad} :

$$1.5 + 1.5(2) = 1 + 3t_{ad}$$
$$4.5 = 1 + 3t_{ad}$$
$$t_{ad} = 7/6$$

Calculando o ponto de intersecção temos (Figura 25):

$$I_{ad} = (x_{máx}, y_{ad}) = (7, 4.5)$$

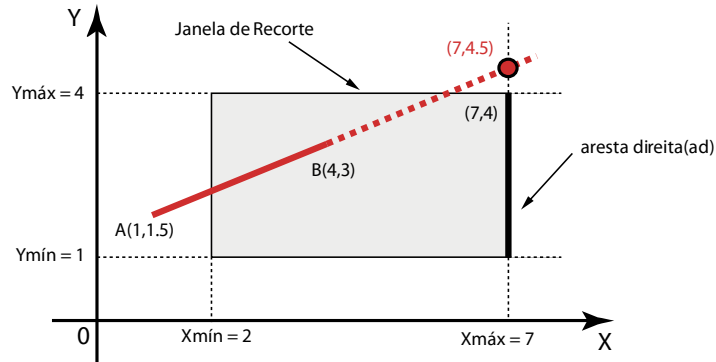


Figura 25: Intersecção de AB com a **aresta direita** (I_{ad}) da **janela de recorte**

O valor de $t_{ad} = 7/6$, indica que o ponto de intersecção não se encontra na aresta direita do retângulo de recorte, pois não está entre 0 e 1. O valor de -2 para t_{AB} indica que o ponto de intersecção não se encontra no interior do seguimento de reta AB, pois t está fora do intervalo $[0,1]$.

Cálculo da intersecção do segmento AB com a aresta esquerda da janela de recorte (I_{ae})

Da aresta esquerda da **janela de recorte** temos:

$$\begin{aligned}x_{ae} &= x_{min} \\y_{ae} &= y_{min} + (y_{máx} - y_{mín})t_{ae}\end{aligned}$$

Substituindo os valores:

$$\begin{aligned}x_{ae} &= 2 + 0 \cdot t_{ae} = 2 \\y_{ae} &= 1 + 3t_{ae}\end{aligned}$$

Como o ponto é de intersecção entre o segmento de reta AB e a **aresta esquerda** da **janela de recorte**, os valores de x e y destes pontos são iguais. Assim:

$$\begin{aligned}x_{AB} &= x_{ae} \\y_{AB} &= y_{ae} \\1 + 3t_{AB} &= 2 \\1.5 + 1.5t_{AB} &= 1 + 3t_{ae}\end{aligned}$$

Temos:

$$t_{AB} = \frac{1}{3}$$

e t_{ae} :

$$1.5 + 1.5\left(\frac{1}{3}\right) = 1 + 3t_{ae}$$

$$t_{ae} = 1/3$$

Calculando o ponto de intersecção:

$$I_{ae} = (x_{mín}, y_{ae}) = (2, 2)$$

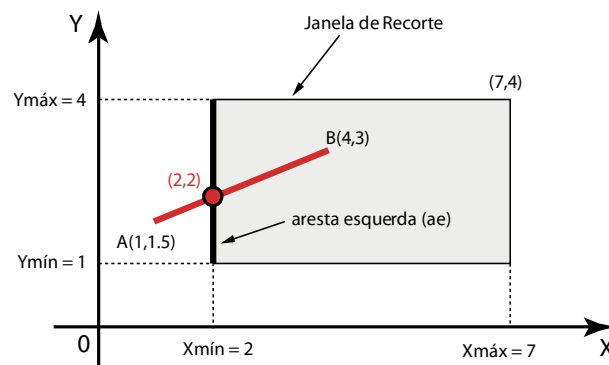


Figura 26: Intersecção de AB com a aresta esquerda (I_{ae}) da janela de recorte

O valor de $t_{ae} = 1/3$, indica que o ponto de intersecção se encontra na aresta esquerda do retângulo de recorte, pois está entre 0 e 1. O valor de $1/3$ para t_{AB} também indica que o ponto de intersecção se encontra no interior do seguimento de reta AB, pois t está dentro do intervalo $[0,1]$, figura 26.

7.1.2.1.2.1 Generalização do Cálculo Paramétrico para Janelas Retangulares

Vimos que em função de t podemos escrever de forma genérica a determinação de um ponto sobre o segmento de reta:

$$P(t_{AB}) = P_A + (P_B - P_A) \cdot t_{AB} ; \quad 0 \leq t_{AB} \leq 1$$

$$t_{AB} = \frac{P(t_{AB}) - P_A}{P_B - P_A}$$

Para encontrar a intersecção com as arestas de uma janela de recorte retangular podemos simplificar os cálculos, pois os limites da janela são conhecidos. Assim o cálculo de t_{AB} visando a intersecção com cada aresta especifica pode ser assim expresso:

Aresta superior (t_{as}):

$$t_{AB} = t_{as} = \frac{y_{\max} - y_A}{y_B - y_A}; \quad 0 \leq t_{as} \leq 1$$

Aresta inferior (t_{ai}):

$$t_{AB} = t_{ai} = \frac{y_{\min} - y_A}{y_B - y_A}; \quad 0 \leq t_{ai} \leq 1$$

Aresta direita (t_{ad}):

$$t_{AB} = t_{ad} = \frac{x_{\max} - x_A}{x_B - x_A}; \quad 0 \leq t_{ad} \leq 1$$

Aresta esquerda (t_{ae}):

$$t_{AB} = t_{ae} = \frac{x_{\min} - x_A}{x_B - x_A}; \quad 0 \leq t_{ae} \leq 1$$

7.1.2.1.3 Síntese sobre o Algoritmo da Força Bruta

Este algoritmo calcula todas as intersecções do segmento de reta com a região de recorte. Desta forma, este procedimento realiza cálculos desnecessários tornando-se muitas vezes ineficiente.

7.1.2.2 Método da Cohen e Sutherland

Cohen e Sutherland desenvolveram um algoritmo que realiza o trabalho de recorte procurando evitar cálculos desnecessários. Ele inicialmente classifica as linhas em **visíveis**, **não visíveis** e as que **possivelmente necessitam ser recortadas**. Para isto, eles definem algumas regiões que uma linha pode ocupar em relação à uma **janela de recorte**, figura 27.

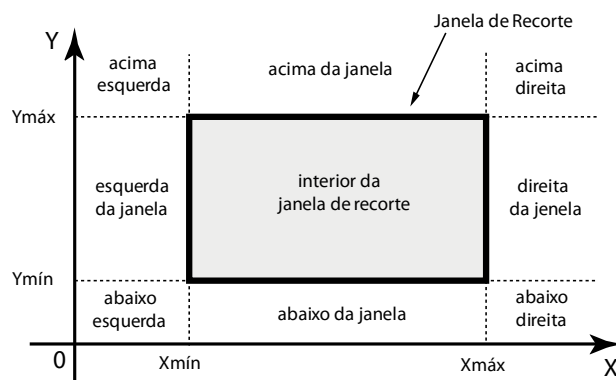


Figura 27: recorte de pontos

Nesta proposta, é preciso identificar em quais regiões se encontram as extremidades dos segmentos de retas. Para facilitar este processo, estas regiões foram representadas por um número de quatro algarismos expresso no sistema binário. Este número binário recebe o nome em inglês de **outcode**, cujos algarismos podem assumir o valor 0 ou 1. Desta forma, foi designado para cada algarismo a representação de uma determinada região que envolve a **janela de recorte** e suas mediações. O primeiro algarismo identifica se o ponto (uma das extremidades do segmento de reta) está acima da **janela de recorte**, o segundo se está abaixo, o terceiro se está à direita e o quarto se está a esquerda (Figura 28).



Figura 28: Outcodes representando regiões que envolvem a **janela de recorte**

Assim, podemos relacionar os algarismos como se fossem bits que podem assumir os valores 0 ou 1. Desta forma, temos a seguinte especificação sobre a localização de um determinado ponto em relação a **janela de recorte**:

Valor **1** para o **Bit 1** = ponto acima da janela

Valor **0** para o **Bit 1** = ponto não se encontra acima da janela

Valor **1** para o **Bit 2** = ponto abaixo da janela

Valor **0** para o **Bit 2** = ponto não se encontra abaixo da janela

Valor **1** para o **Bit 3** = ponto a direita da janela

Valor **0** para o **Bit 3** = ponto não se encontra a direita da janela

Valor **1** para o **Bit 4** = ponto a esquerda da janela

Valor **0** para o **Bit 4** = ponto não se encontra a esquerda da janela

Obs: quando todos os bits forem 0, ou seja, com **outcode = 0000**, este conjunto de bits indica que o ponto está no interior da **janela de recorte**. Desta forma, as regiões recebem **outcodes** correspondentes a sua posição em relação a janela de visualização. A figura 29 mostra os **outcodes** para cada uma delas.

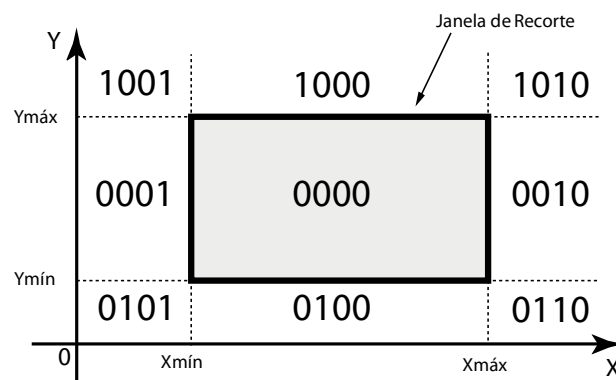


Figura 29: outcodes de regiões que envolvem a janela

Um segmento de reta pode ser definida por dois pontos distintos no espaço. Assim, dependendo da posição destes pontos, poderemos identificar se uma determinada reta ou segmento de reta será visível ou não, bem como se necessita ser recortada.

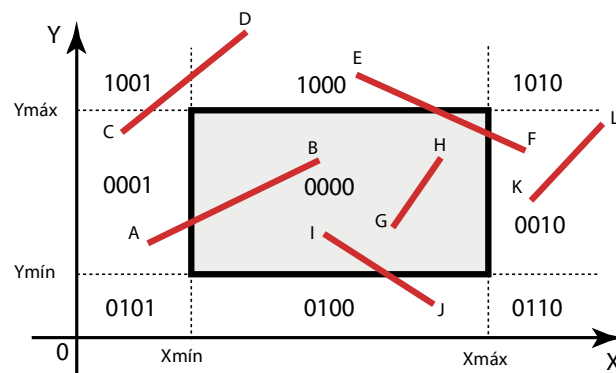


Figura 30: recorte de pontos

Para ilustrar o funcionamento desta proposta tomemos a figura 29. Cada segmento de reta é definido por dois pontos distintos no plano. Assim, para cada reta temos dois pontos a ser avaliados da seguinte forma:

Tabela 1 – Exemplo de **outcodes** para a Figura 30

Segmento	Outcode para o Primeiro Ponto	Outcode para o Segundo Ponto
AB	A: 0001	B: 0000
CD	C: 0001	D: 1000
EF	E: 1000	F: 0010
GH	G: 0000	H: 0000
IJ	I: 0000	J: 0100
KL	K: 0010	L: 0010

Agora, precisamos saber quais pontos precisam ser recortados e onde estes **outcodes** nos ajudam nesta tarefa. **Cohem e Sutherland**, analisando os **outcodes** dos pontos que definem um segmento de reta e as possíveis localizações que ele pode assumir, definiram:

1. **O segmento será visível:** se os dois pontos extremos tiverem **outcodes 0000**, está reta se encontra na interior da **janela de recorte**. Ambos devem ter o mesmo **outcode = 0000**.
2. **O segmento não será visível na janela:** se fizermos a associação lógica bit a bit do ponto inicial com o ponto final do segmento de reta e, desta associação, os bits forem iguais, mantemos para este bit o valor, caso haja diferença nos valores, será atribuído o valor **0**. Ao final deste procedimento se o resultado desta operação (**AND lógico**) resultar em um **outcode** diferente de **0000**, este segmento estará fora do retângulo de recorte.
3. **Parte do segmento pode ser visível:** Considerando a análise feita no item 2, se **outcode** for **0000**, o segmento deverá ser recortado, pois, é possível que parte dele esteja no interior da **janela de recorte**.

Aplicando este raciocínio para os segmentos mostrados na figura 30 temos a seguinte classificação:

Tabela 2: – Resultado dos *outcodes* para a Figura 30

Segmento	Primeiro ponto	Segundo ponto	Outcode	Resultado
AB	A: 0001	B: 0000	0000	Recorte
CD	C: 0001	D: 1000	0000	Recorte
EF	E: 1000	F: 0010	0000	Recorte
GH	G: 0000	H: 0000	0000	Visível
IJ	I: 0000	J: 0100	0000	Recorte
KL	K: 0010	L: 0010	0010	Não Visível

Vamos apresentar alguns exemplos para ilustrar como funciona o método proposto por **Cohen e Sutherland**, para depois, construir um algoritmo.

Exemplo-03: Tomemos a figura 31 que descreve uma janela de visualização com seus limites e um determinado segmento de reta definido pelas coordenadas dos pontos **A** e **B**.

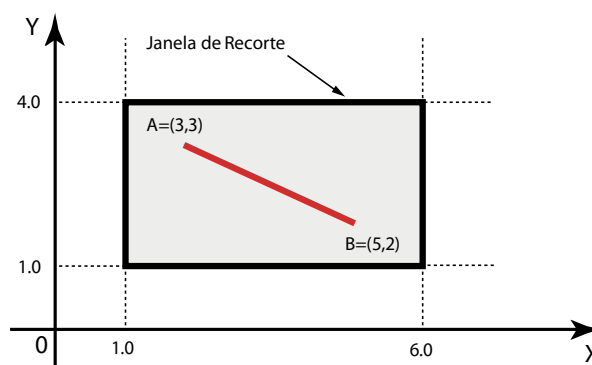


Figura 31: Segmento de reta AB no interior da **janela de recorte**

Vamos analisar os extremos do segmento utilizando a metodologia proposta por **Cohen-Sutherland**. Temos o ponto **A(3,3)** e o ponto **B(5,2)**. Inicialmente vamos determinar os outcodes destes pontos.

Ponto A(3,3) = **0000**

Ponto B(5,2) = **0000**

O **primeiro passo** é verificar se ambos os outcodes são **0000**. Se ambos satisfizerem esta condição podemos afirmar que estão contidos no interior da **janela de recorte**. Analisando os **outcodes** temos que ambos são **0000**, portando, este teste indica que o segmento **AB** está totalmente dentro da **janela de recorte**, ou seja, **visível**. Aceita esta condição, nenhum outro teste precisa ser realizado, apenas desenhar o segmento.

Exemplo 04: Neste segundo exemplo vamos analisar o segmento de reta **AB** mostrado na figura 32 e aplicar a metodologia apresentada por **Cohen e Sutherland**.

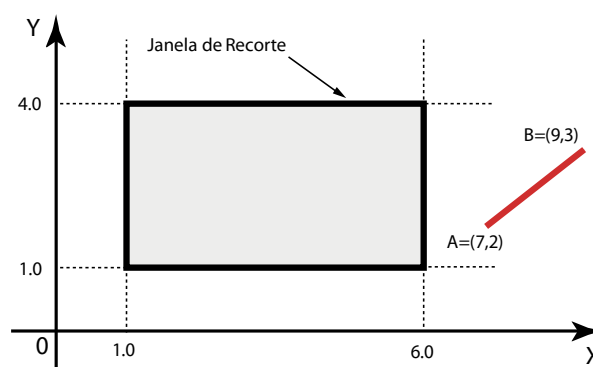


Figura 32: Segmento de reta AB fora da **janela de recorte**

Mais uma vez vamos analisar os extremos do segmento utilizando a metodologia proposta por **Cohen-Sutherland**. Temos o pontos **A(7,2)** e o ponto **B(9,3)**. Inicialmente vamos determinar os **outcodes** destes pontos.

Ponto A(7,2) = **0010**

Ponto B(9,3) = **0010**

Vimos que o **primeiro passo** e testar se o dois pontos simultaneamente possuem **outcodes 0000**. Neste caso podemos observar que eles não são **0000**. Desta forma, em que os **outcodes** não são simultaneamente iguais a **0000**, precisamos realizar um **segundo passo**. Nesta etapa precisamos realizar o **AND lógico** entre os **outcodes** dos pontos A e B. Assim, temos:

Ponto A **0010**

Ponto B **0010**

AND AB **0010**

Podemos notar que comparados os bits temos que eles são iguais. O **AND lógico** nos ensina que se os bits forem iguais eles serão repetidos e se forem diferentes assumirão o valor 0. Assim, temos como resultado **0010**. Que é diferente de **0000**. Deste resultado, podemos concluir que o segmento de reta está totalmente fora da **janela de recorte**.

Exemplo 05: Neste exemplo temos um segmento de reta parcialmente dentro da **janela de recorte** ocupando mais de uma região (Figura 33). Vamos analisar o comportamento do método proposto por **Cohen e Sutherland** nesta situação.

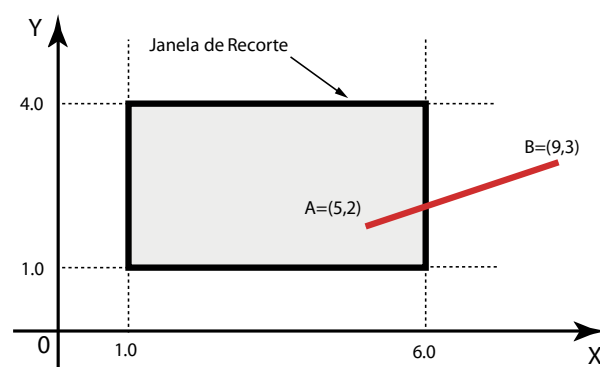


Figura 33: Segmento de reta AB parcialmente dentro da **janela de recorte**

Iniciamos determinando os **outcodes** dos pontos **A** e **B** do segmento de reta. Temos **A(5,2)** e **B(9,3)**.

Ponto A(5,2) = **0000**

Ponto B(9,3) = **0010**

Vimos que o **primeiro passo** é testar se os dois pontos simultaneamente possuem **outcodes 0000**. Neste caso, podemos observar que eles não são simultaneamente **0000**. Precisamos então realizar um **segundo passo** que consiste em testar bit a bit os **outcodes** de A e de B de forma que se os valores contidos nos bits forem iguais, o resultado é o valor coincidente encontrado, caso contrário, o resultado será **0** (AND lógico).

Ponto A = **0000**

Ponto B = **0010**

AND AB = **0000**

Neste exemplo, o **AND lógico** produziu **0000**. Este resultado indica que o segmento deverá ser recortado, pois pode ser visível. Assim, vamos calcular a intersecções do segmento com a **janela de recorte** para executar o recorte.

Da janela temos:

$$x_{\min} = 1.0$$

$$y_{\min} = 1.0$$

$$x_{\max} = 6.0$$

$$y_{\max} = 4.0$$

Do segmento de reta:

$$x_A = 5.0$$

$$y_A = 2.0$$

$$x_B = 9.0$$

$$y_B = 3.0$$

As equações paramétricas da reta AB:

$$x_{AB} = x_A + (x_B - x_A)t_{AB}$$

$$y_{AB} = y_A + (y_B - y_A)t_{AB}$$

$$x_{AB} = 5 + 4t_{AB}$$

$$y_{AB} = 2 + t_{AB}$$

O Ponto **A** do segmento de reta **AB** possui **outcode 0000** e o ponto **B 0010**. Note que precisamos ter os dois **outcodes** com **0000**. Assim, vamos fazer o recorte com a **aresta direita** da **janela de recorte**, pois, o primeiro bit diferente de 0 é o terceiro bit, que por sua vez, indica que o ponto **B** está na região a direita da **janela de recorte**.

Desta forma, vamos fazer o cálculo da inteseccção do segmento AB com a aresta direita da janela de recorte (I_{ad}). Da aresta direita da **janela de recorte** temos:

$$x_{ad} = x_{\max}$$

$$y_{ad} = y_{\min} + (y_{\max} - y_{\min})t_{ad}$$

Substituindo os valores:

$$x_{ad} = 6$$
$$y_{ad} = 1 + 3t_{ad}$$

Como o ponto é de intersecção entre o segmento de reta AB e a **aresta direita da janela de recorte**, os valores de x e y destes pontos são iguais. Temos:

$$x_{AB} = x_{ad}$$

$$y_{AB} = y_{ad}$$

$$5 + 4t_{AB} = 6$$

$$2 + t_{AB} = 1 + 3t_{ad}$$

e:

$$t_{AB} = 1/4$$

Para t_{ad} :

$$2 + 1/4 = 1 + 3t_{ad}$$

$$2.25 = 1 + 3t_{ad}$$

$$t_{ad} = 5/12$$

Calculando o ponto de intersecção temos (Figura 34):

$$I_{ad} = (x_{máx}, y_{ad}) = (6, 2.25)$$

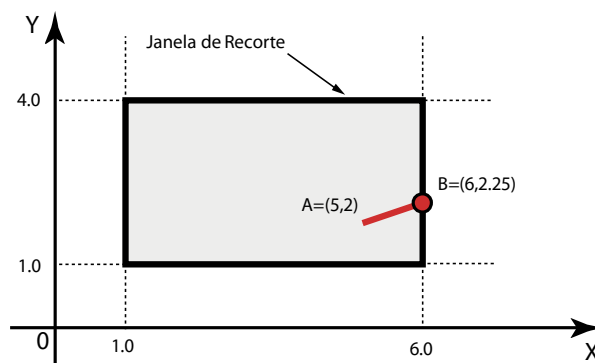


Figura 34: Intersecção de AB com a **aresta direita** (I_{ad}) da **janela de recorte**

O valor de $t_{ad} = 5/12$, indica que o ponto de intersecção se encontra na aresta direita do retângulo de recorte, pois está entre 0 e 1. O valor de $\frac{1}{4}$ para t_{AB} indica que o ponto de intersecção se encontra no interior do seguimento de reta AB, pois t está dentro do intervalo $[0,1]$.

Temos agora que testar os **outcodes** de **A** e **I_{ad}** começando do passo 1.

Ponto A = **0000**

Ponto I_{ad} = **0000**

Como ambos **outcodes** são iguais a **0000** temos que o segmento recortado **AI_{ad}** encontra-se no interior da **janela de recorte**.

Exemplo 06: Neste exemplo temos um segmento de reta fora da **janela de recorte**, ocupando mais de uma região (Figura 35). Analogamente ao exemplo anterior, vamos analisar o comportamento do método proposto por **Cohen e Sutherland**.

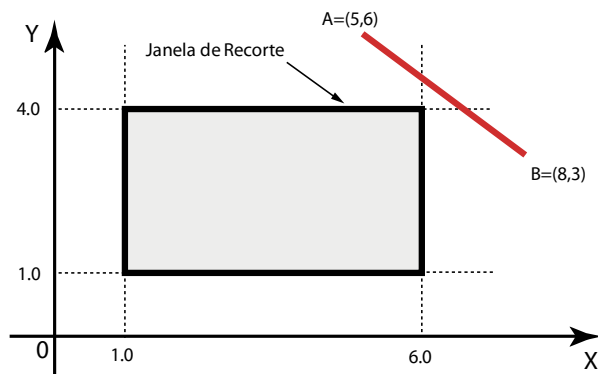


Figura 35: Segmento de reta AB fora da **janela de recorte** ocupando diferentes regiões

Iniciamos determinando os **outcodes** dos pontos **A** e **B** do segmento de reta. Temos **A(5,6)** e **B(8,3)**.

Ponto A(5,6) = **1000**

Ponto B(8,3) = **0010**

Como nos outros exemplos, o **primeiro passo** é testar se o dois pontos simultaneamente possuem **outcodes 0000**. Neste caso, podemos observar que eles não são simultaneamente **0000**. Desta forma, precisamos realizar o **segundo passo**, que consiste

em testar bit a bit os **outcodes** de A e de B de forma que se os valores contidos nos bits forem iguais, o resultado é o valor coincidente encontrado nos bits, caso contrário, o resultado será **0 (AND lógico)**.

Ponto A = **1000**

Ponto B = **0010**

AND AB = **0000**

Neste exemplo, o **AND lógico** produziu **0000**. Este resultado, durante o segundo passo, indica que o segmento deverá ser recortado, pois pode ser visível. Vamos então calcular as possíveis intersecções do segmento **AB** com a **janela de recorte** para executar o recorte.

Da janela temos:

$$x_{\min} = 1.0$$

$$y_{\min} = 1.0$$

$$x_{\max} = 6.0$$

$$y_{\max} = 4.0$$

Do segmento de reta:

$$x_A = 5.0$$

$$y_A = 6.0$$

$$x_B = 8.0$$

$$y_B = 3.0$$

Para a equação da reta AB:

$$x_{AB} = x_A + (x_B - x_A)t_{AB}$$

$$y_{AB} = y_A + (y_B - y_A)t_{AB}$$

$$x_{AB} = 5 + 3t_{AB}$$

$$y_{AB} = 6 - 3t_{AB}$$

Para determinar sobre qual aresta da janela de recorte iremos calcular a intersecção com o segmento de reta **AB**, vamos observar os outcodes dos pontos **A** e **B** e verificar em quais posições os bits indicam o valor **1**. Iniciando a observação com o ponto **A** (outucode **1000**) notamos que o primeiro BIT possui valor **1**. Este Bit indica que o ponto esta acima da **janela**

de recorte, portando calcularemos o ponto de **inteseccção com a aresta superior** (I_{as}) desta janela.

Da aresta superior da **janela de recorte** temos:

$$\begin{aligned}x_{as} &= x_{min} + (x_{m\acute{a}x} - x_{min})t_{as} \\y_{as} &= y_{m\acute{a}x}\end{aligned}$$

Substituindo os valores:

$$\begin{aligned}x_{as} &= 1 + 5t_{as} \\y_{as} &= 4\end{aligned}$$

Como o ponto é de intersecção entre o segmento de reta AB e a **aresta superior da janela de recorte**, os valores de x e y destes pontos são iguais. Assim:

$$\begin{aligned}x_{AB} &= x_{as} \\y_{AB} &= y_{as}\end{aligned}$$

$$\begin{aligned}5 + 3t_{AB} &= 1 + 5t_{as} \\6 - 3t_{AB} &= 4\end{aligned}$$

Calculando t_{AB} :

$$t_{AB} = 2/3$$

Para t_{as} temos:

$$\begin{aligned}5 + 3 \cdot \left(\frac{2}{3}\right) &= 1 + 5t_{as} \\t_{as} &= 6/5 \\I_{as} &= (7,4)\end{aligned}$$

Precisamos avaliar agora os segmento AI_{as} e o segmento $I_{as}B$ (figura 36). De acordo com a metodologia proposta por **Cohen e Sutherland** temos:

$$\begin{aligned}\text{Ponto A} &= \mathbf{1000} \\ \text{Ponto } AI_{as} &= \mathbf{1010} \\ \text{AND Lógico} &= \mathbf{1000}\end{aligned}$$

Observe que o AND lógico é diferente de 0000. Este resultado indica que o segmento AI_{as} Não será visível. Temos que agora avaliar o seguimento $I_{as}B$.

Ponto AI_{as} = 1010

Ponto B = 0010

AND Lógico = 0010

Como o resultado do **AND lógico** não é 0000, o segmento $I_{as}B$ também não será visível.

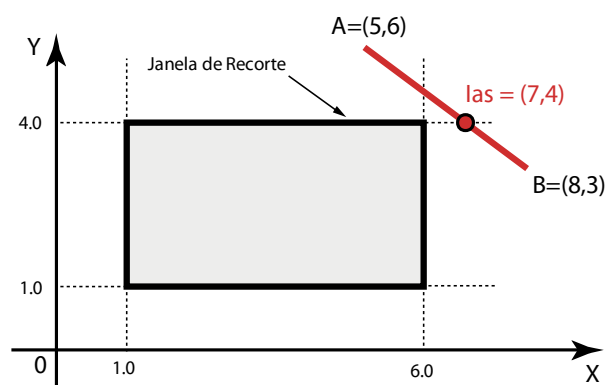


Figura 36: aplicação do algoritmo de Cohen-Sutherland em um segmento fora da janela de visualização

7.1.2.2.1 Síntese sobre o Algoritmo de Cohen e Sutherland

Este algoritmo detecta facilmente segmentos de reta que estejam totalmente no interior ou no exterior da janela de recorte evitando cálculos desnecessários (estes casos são conhecidos na literatura como aceitação trivial e rejeição trivial respectivamente). Além disso, facilita a identificação da aresta da janela de recorte contra a qual se deve recortar a linha. Quando se tem um volume grande de segmentos nestas condições o algoritmo se mostra muito proveitoso. No entanto, quando se tem uma quantidade expressiva de segmentos que não são enquadrados como diretamente dentro ou fora da **janela de recorte**, este algoritmo realiza cálculos de intersecções determinando pontos que estão fora da janela de recorte, que devido a esta característica, deveriam ser evitados. Além disso, este algoritmo é destinado apenas a regiões de recorte retangulares.

7.1.2.3 Algoritmo de Cyrus-Beck

Este algoritmo utiliza a equação paramétrica da reta para determinar as possíveis inteseções de um segmento de reta com as arestas de um polígono convexo de recorte. Observe que ele pode ser utilizado não apenas em regiões retangulares de recorte (figura 37).

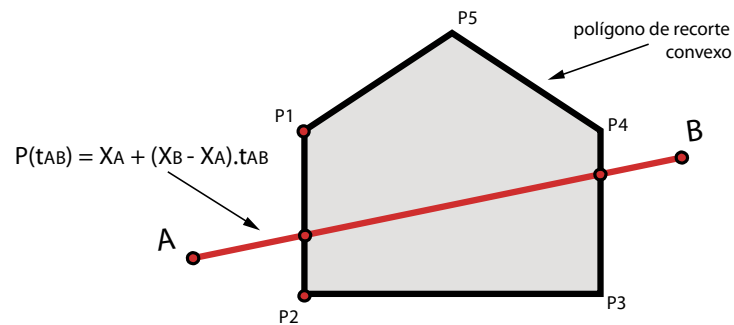


Figura 37: Segmento de reta AB e um polígono de recorte convexo

O Algoritmo de Mark Cyrus e Jay Beck se fundamenta em determinar a inteseção do segmento de reta com as arestas do polígono de recorte valendo-se do produto interno entre o vetor normal (**N**) de cada aresta e um vetor formado por um ponto pertencente a aresta e o segmento de reta (**P1P(t_{AB})**), figura 38.

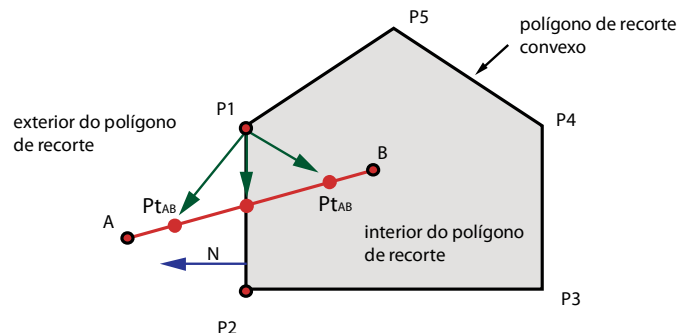


Figura 38: Segmento de reta AB e um polígono de recorte convexo

O produto interno entre os vetores **N** e **P1P(t_{AB})** pode apresentar três possibilidades em termos de sinais (figura 39):

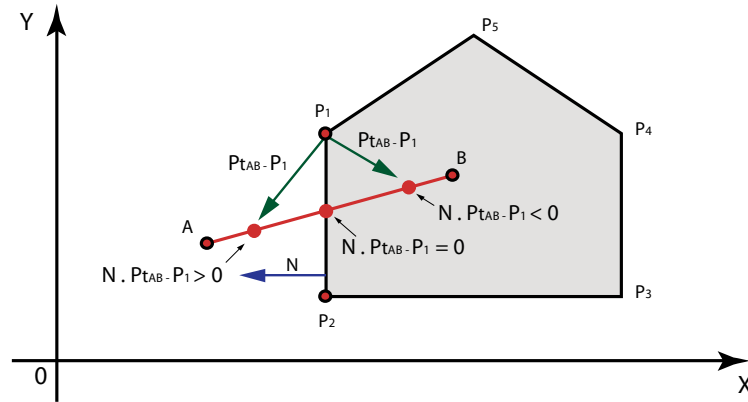


Figura 39: O produto interno entre os vetores \mathbf{N} e $\mathbf{P1P(t_{AB})}$

$\mathbf{N} \cdot \mathbf{P_1P_{t_{AB}}} = 0$: se os vetores forem perpendiculares (ângulo entre eles é de 90°);

$\mathbf{N} \cdot \mathbf{P_1P_{t_{AB}}} > 0$: se o ângulo formado entre os dois vetores for menor que 90° ;

$\mathbf{N} \cdot \mathbf{P_1P_{t_{AB}}} < 0$: se o ângulo formado entre os dois vetores for maior que 90° .

As equação paramétrica da reta \mathbf{AB} pode ser escrita da seguinte forma:

$$x_{AB} = x_A + (x_B - x_A)t_{AB}$$

$$y_{AB} = y_A + (y_B - y_A)t_{AB}$$

Genericamente:

$$\mathbf{P}(t_{AB}) = \mathbf{P}_A + (\mathbf{P}_B - \mathbf{P}_A)t_{AB}$$

Lembrando sempre que t_{AB} deve estar no intervalo de $[0,1]$ o que implica que $\mathbf{P}(0) = \mathbf{P}_A$ e $\mathbf{P}(1) = \mathbf{P}_B$, vamos calcular o produto interno entre \mathbf{N} e $\mathbf{P1P(t_{AB})}$ quando este vetor for coincidente, no caso da figura 39, com a aresta $\mathbf{P1P2}$. Neste caso ($\mathbf{P1P2}$), é naturalmente, normal à referida aresta. Quando isto ocorre, temos o valor de t_{AB} que permite o cálculo da intersecção do segmento de reta \mathbf{AB} com a aresta $\mathbf{P1P2}$ do polígono convexo de recorte. Vamos ao cálculo deste produto interno.

$$\mathbf{N} \cdot \mathbf{P_1P_{t_{AB}}} = 0$$

Como $\mathbf{P_1P_{t_{AB}}} = \mathbf{P(t_{AB})} - \mathbf{P_1}$, temos:

$$\mathbf{N} \cdot (\mathbf{P(t_{AB})} - \mathbf{P_1}) = 0$$

Substituindo $P(t_{AB})$ na equação acima:

$$N \cdot (P_A + (P_B - P_A)t_{AB} - P_1) = 0$$

$$N \cdot (P_A - P_1) + N \cdot (P_B - P_A) \cdot t_{AB} = 0$$

Isolando t_{AB} e fazendo $D = P_B - P_A$:

$$t_{AB} = \frac{N \cdot (P_A - P_1)}{-N \cdot D}$$

Observe que o denominador da equação acima precisa ser diferente de zero. O que implica nas seguintes condições:

$N \neq 0$: garante que o vetor normal não pode ser nulo;

$D \neq 0$: garante que temos um segmento de reta ($P_A \neq P_B$);

$N \cdot D \neq 0$: garante que a aresta e o segmento não são paralelos. Caso contrário não temos intersecção.

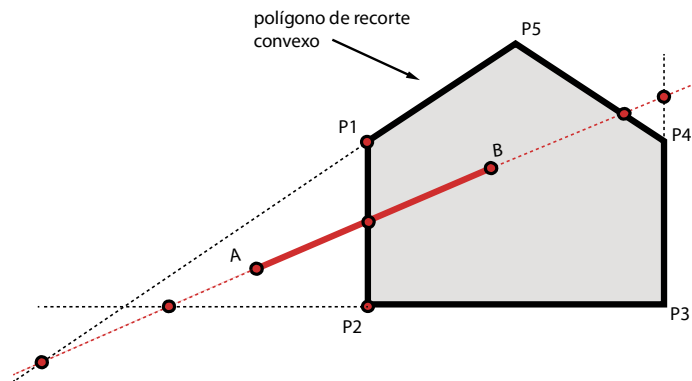


Figura 40: Apenas a intersecção com P1P2 interessa

Observe que com os pontos que determinam o segmento de reta **AB** e os que definem as arestas do polígono de recorte podemos determinar os pontos de intersecção do segmento de reta **AB** com todas as arestas do polígono. No entanto, precisamos somente daquelas intersecções necessárias para a execução do recorte. Desta forma, torna-se interessante calcular apenas as intersecções efetivas com a janela de recorte. Note, por exemplo, que na figura 36, embora possamos calcular as intersecções do segmento de reta **AB** com todas as

arestas do polígono de recorte, apenas a intersecção com a aresta **P1P2** nos interessa (figura 40).

Temos que evitar cálculos desnecessários. Neste sentido, o algoritmo de **Mike Cyrus e Jay Beck** (1978) propõe a avaliação do sinal do produto interno entre os vetores formados pelas arestas do polígono de recorte e o segmento de reta a ser recortado, ou seja **AB** (figura 41).

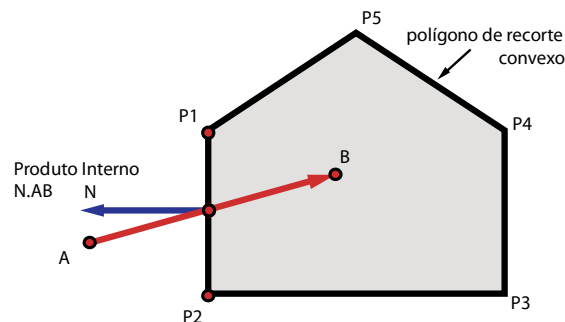


Figura 41: O produto interno entre os vetores **N** e uma aresta do polígono de recorte

O sinal do produto interno entre os vetores normais (**N**) das arestas do polígono de recorte e o segmento de reta **AB** depende dos sentidos adotados para os respectivos vetores. Assim, é preciso que a numeração dos vértices do polígono seja feita adotando-se uma determinada sequência lógica. Aqui, adotamos uma sequência contrária ao movimento dos ponteiros do relógio (anti-horária), como pode ser observado na figura 41. Além disso, também adotamos que os vetores normais as arestas, apontarão para o exterior do polígono de recorte.

Observando a figura 41 notamos que o segmento **AB** aponta de **A** para **B**. Isto ocorre porque a construção do vetor **AB** se dá subtraindo-se do ponto final, o inicial, ou seja: **B-A**. Com estas informações, vamos analisar o sinal produzido pelo produto interno entre **AB** e uma determinada aresta do polígono, no caso, **P1P2**. Assim, de forma genérica, podemos notar que, se o produto interno for positivo, indica que o ângulo entre o vetor **N** e o segmento **AB** é menor que 90° , zero se for igual e negativo se for maior que 90° (figura 42).

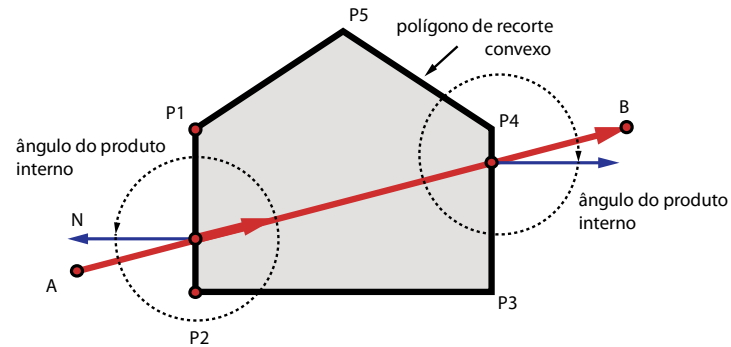


Figura 42: O produto interno entre os vetores **N** das arestas **P1P2** e **P3P4** e o segmento **AB**

Perceba que no caso da aresta **P1P2** o produto interno entre o vetor normal desta aresta e o vetor produzido por **AB** é negativo, pois o ângulo é maior que 90° . Já o produto interno com o vetor normal da aresta **P3P4** é positivo, pois neste caso, o ângulo entre os vetores é menor que 90° , portanto, positivo. Observe que no caso da aresta **P1P2** o segmento **AB** pode ser visto como que entrando no **polígono de recorte** e por essa razão, quando o produto interno for negativo, o convencionaremos, como ponto de entrada. Caso contrário, isto é, quando o produto interno for positivo, o rotularemos como ponto de saída. Em síntese temos:

$$N \cdot P_A P_B < 0 \text{ Ponto de Entrada}$$

$$N \cdot P_A P_B > 0 \text{ Ponto de Saída}$$

Observando a figura 36 notamos que o polígono de recorte possui 5 arestas. Fato que nos leva a calcular 5 valores de t_{AB} , um para cada aresta. No entanto, os valores que estão fora do intervalo $[0,1]$ devem ser descartados (figura 43).

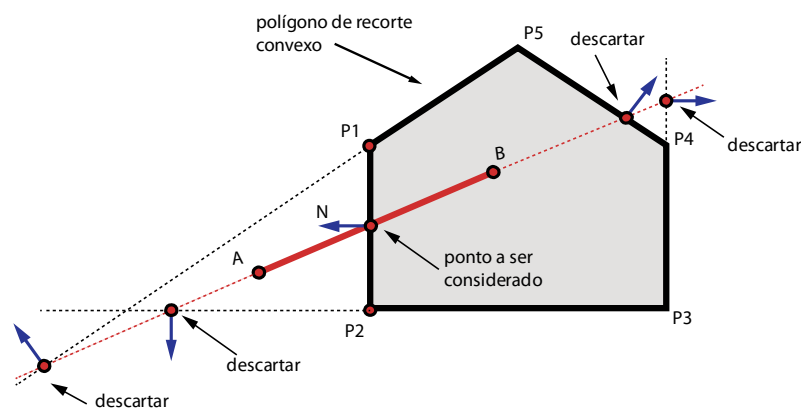


Figura 43: Apenas o ponto de interseção do segmento **AB** com a aresta **P1P2** possui t_{AB} no intervalo $[0,1]$

Além disso, podemos ter casos em que os segmentos de retas a serem recortados cruzam duas arestas. O algoritmo de Cyrus-Beck evita o cálculo de intersecções que não são pertinentes ao recorte. Isto pode ser feito classificando, de acordo com o valor do produto interno entre a respectiva aresta e o segmento AB, os valores de t_{AB} como de entrada ou de saída. Da figura 44 podemos notar que pontos de intersecção de entrada, ou seja, com produto interno negativo, são maiores, quanto mais próximos estiverem da intersecção.

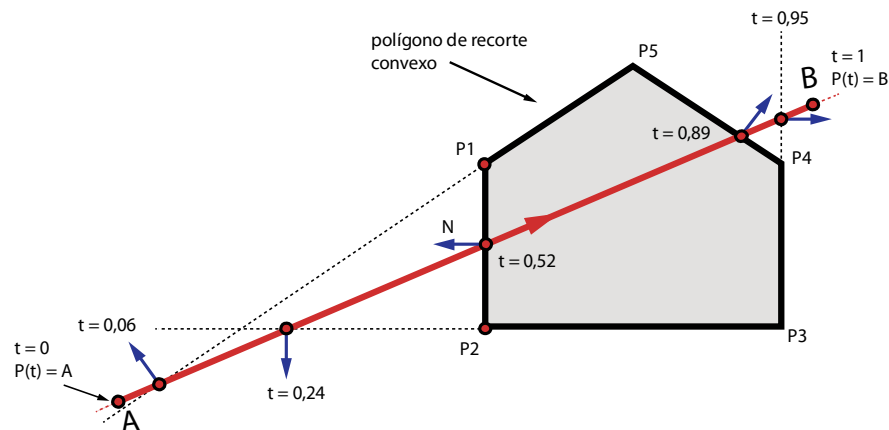


Figura 44: Valores de t_{AB} crescem entre $[0,1]$ de A para B no segmento de reta que possui sentido AB

Desta forma, o algoritmo manda adotar o maior valor de t_{AB} , entre os de entrada que estiverem entre $[0,1]$. De forma análoga, manda adotar o menor valor de t_{AB} , entre os rotulados como de saída que estiverem entre $[0,1]$. Com estes valores de t_{AB} determinados, calculamos os pontos de intersecção necessários ao recorte. Um exemplo pode deixar mais claro o funcionamento do algoritmo.

Algoritmo de Cyrus-Beck para o Recorte de Segmentos de Reta

- **Passo 1:** determinar os valores de t_{AB} relativos as intersecções do segmento de reta **AB** com as arestas do polígono de recorte;
- **Passo 2:** determinar o produto interno entre o vetor normal de cada aresta e o segmento de reta **AB**;
- **Passo 3:** classificar os valores de t_{AB} de cada aresta como de entrada ou de saída;
- **Passo 4:** ordenar os valores de t_{AB} de entrada e os de saída;
- **Passo 5:** eleger para cálculo dos pontos de intersecção o maior dos

valores de t_{AB} entre os de entrada e o menor entre os de saída que estiverem entre $[0,1]$;

- **Passo 6:** calcular os pontos de intersecção
- **Passo 7:** traçar o segmento recortado.

```
{
  Calcule Ni e escolha um PEi para cada aresta

  tE = 0;
  tL = 1;
  for (cada aresta ){
    if (Ni.(P1-P0)!=0 ){ /* aresta não é paralela ao segmento */
      calcule t;
      use sign of Ni.(P1-P0) para categorizar como PE ou PL;
      if( PE ) tE = max(tE, t);
      if( PL ) tL = min(tL, t);
    } else { /* aresta paralela ao segmento */
      if (Ni.(P0-PEi) > 0) /* está fora */
        return nil;
    }
    if (tE > tL)
      return nil;
    else
      return P(tE) and P(tL) as true clip intersections;
  }
}
```

Exemplo 07: Vamos considerar um polígono triangular como janela de recorte com coordenadas **P1(6,1)**, **B(9,6)** e **C(6,4)** e um segmento de reta definido pelo pontos **A(2,1)** e **B(10,4)** e aplicar o algoritmos de Cyrus-Beck que pode ser resumido nos seguintes passos (figura 45):

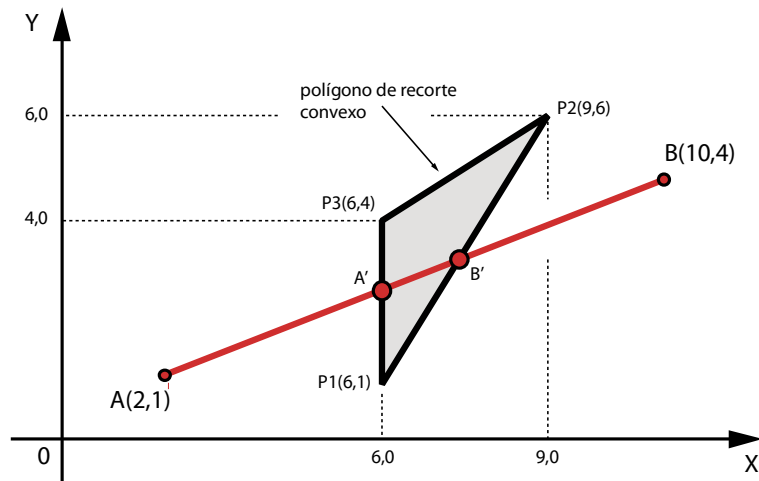


Figura 45: Apenas o ponto de intersecção do segmento AB com a aresta P1P2 possui t_{AB} no intervalo $[0,1]$

Passo 1:

Sabemos que o cálculo de t_{AB} para cada aresta do polígono de recorte é dado por :

$$t_{AB} = \frac{N \cdot (P_A - P_n)}{-N \cdot D}$$

Sendo:

$$D = P_B - P_A$$

$$D = (10 - 2)\vec{i} + (4 - 1)\vec{j}$$

$$D = 8\vec{i} + 3\vec{j}$$

Cálculo do vetor normal para cada aresta:

Vamos ilustrar como pode ser feito o cálculo do vetor normal a um determinado vetor. Tomemos os pontos P1 e P2 e vamos calcular o vetor normal a P1P2. Sabemos que os vetores são perpendiculares. Desta forma temos que:

$$N \cdot P1P2 = 0$$

Tomando N como:

$$N = a\vec{i} + b\vec{j}$$

e

$$P1P2 = (x_{P2} - x_{P1})\vec{i} + (y_{P2} - y_{P1})\vec{j}$$

$$P1P2 = 3\vec{i} + 5\vec{j}$$

$$N.P1P2 = (a\vec{i} + b\vec{j}).(3\vec{i} + 5\vec{j}) = 0$$

$$3a + 5b = 0$$

Esta equação pode ser satisfeita com vários valores de **a** e **b**. Desta forma, vamos adotar um valor para **a** e calcular **b**. Adotando o valor 1 para **a** temos o seguinte valor para **b**:

$$3.1 + 5b = 0$$

$$b = -\frac{3}{5}$$

Para facilitar os cálculos vamos multiplicar ambos os valores de **a** e **b** por 5. Assim, obtemos os seguintes valores de **a** e **b**: **a = 5** e **b = 3**.

$$P1P2 = 3\vec{i} + 5\vec{j}$$

$$N_{P1P2} = 5\vec{i} - 3\vec{j}$$

$$P2P3 = -3\vec{i} - 2\vec{j}$$

$$N_{P2P3} = -2\vec{i} + 3\vec{j}$$

$$P3P1 = 0\vec{i} - 3\vec{j}$$

$$N_{P3P1} = -3\vec{i} + 0\vec{j}$$

Para cada aresta temos PA-Pn:

$$PA - P1 = -4\vec{i} + 0\vec{j}$$

$$PA - P2 = -7\vec{i} - 5\vec{j}$$

$$PA - P3 = -4\vec{i} - 3\vec{j}$$

Para cada aresta temos os seguintes vetores normais:

$$N_{P1P2}.PAP1 = -20$$

$$N_{P2P3}.PAP2 = -1$$

$$N_{P3P1}.PAP3 = 12$$

Podemos calcular o t de intersecção com cada aresta:

$$t_{P1P2} = 0.645$$

$$t_{P2P3} = -0,143$$

$$t_{P3P1} = 0.50$$

Passo 2:

Vamos calcular o produto interno entre as normais e o vetor formado pelo segmento de reta **AB**.

$$N_{P_1P_2} \cdot AB = (5\vec{i} - 3\vec{j}) \cdot (8\vec{i} + 3\vec{j}) = 31 \text{ ponto de saída}$$

$$N_{P_2P_3} \cdot AB = (-2\vec{i} + 3\vec{j}) \cdot (8\vec{i} + 3\vec{j}) = -7 \text{ ponto de entrada}$$

$$N_{P_3P_1} \cdot AB = (-3\vec{i} + 0\vec{j}) \cdot (8\vec{i} + 3\vec{j}) = -24 \text{ ponto de entrada}$$

Passo 3:

Tabela 1: cálculo do parâmetro t de entrada e saída

Aresta	<i>t para pontos de entrada</i>	<i>t para pontos de saída</i>
P1P2		0.645
P2P3	-0.143	
P3P1	0.500	

Passo 4:

Tabela 2: classificação de pontos de entrada

Aresta	<i>maior t de entrada</i>
P2P3	-0.143
P3P1	0.50

Tabela3: classificação de pontos de saída

Aresta	<i>menor t de saída</i>
P1P2	0.645

Passo 5:

Temos como maior t de entrada que esteja no intervalo de [0,1] o valor **0.5**. Como menor valor de t de saída no intervalo de [0,1] temos o valor **0.645**. Portanto, chegamos ao intervalo de t **[0,5, 0,645]**.

Passo 6:

A tabela 1 traz os resultados da aplicação do algoritmo de **Cyrus Beck** para este exemplo e dela temos que escolher o maior t_{AB} de entrada que esteja no intervalo de $[0,1]$ e o menor de saída que esteja também no intervalo $[0,1]$. Assim, temos os valores **0,50 e 0,645** que aplicados a equação paramétrica da reta temos o seguinte segmento de reta:

$$x_{A'} = x_A + (x_B - x_A)t_{AB}$$

$$y_{A'} = y_A + (y_B - y_A)t_{AB}$$

Para $t_{AB} = 0.50$:

$$x_{A'} = x_{iAB} = 2 + (10 - 2)0.50 = 6$$

$$y_{A'} = y_{iAB} = 1 + (4 - 1)0.50 = 2.5$$

Para $t_{AB} = 0.645$:

$$x_{B'} = x_{iAB} = 2 + (10 - 2)0.645 = 7.16$$

$$y_{B'} = y_{iAB} = 1 + (4 - 1)0.645 = 2.94$$

O que nos leva ao segmento de **A'(6, 2.5) a B'(7.16, 2.5)**, figura (46).

Passo 7:

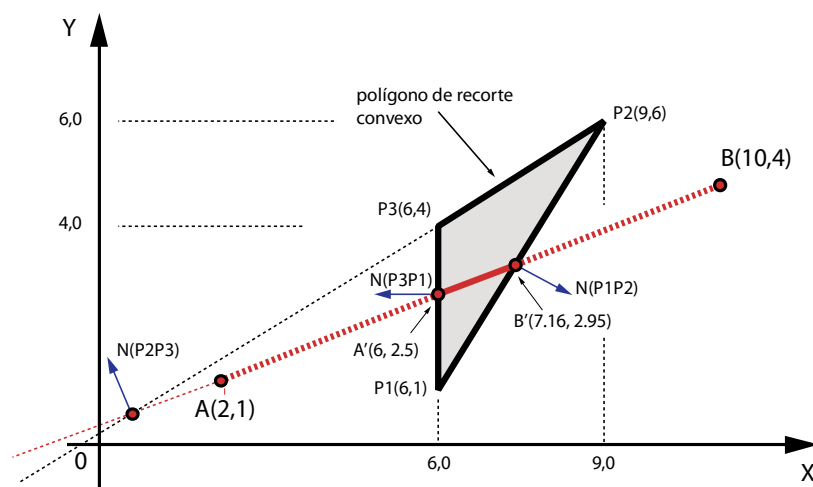


Figura 46: gráfico do resultado do recorte A'B'

Exemplo 8: Vamos considerar um retângulo de recorte com coordenadas **P1(-3,1)**, **B(3,1)**, **C(3,5)** e **(-3,5)** e um segmento de reta definido pelo pontos **A(-4,2)** e **B(-1,6)** e aplicar o algoritmos de Cyrus-Beck que pode ser resumido nos seguintes passos (figura 47):

- **Passo 1:** determinar os valores de t_{AB} relativos as intersecções do segmento de reta **AB** com as arestas do polígono de recorte;
- **Passo 2:** determinar o produto interno entre o vetor normal de cada aresta e o segmento de reta **AB**;
- **Passo 3:** classificar os valores de t_{AB} de cada aresta como de entrada ou de saída;
- **Passo 4:** ordenar os valores de t_{AB} de entrada e os de saída;
- **Passo 5:** eleger para cálculo dos pontos de intersecção o maior dos valores de t_{AB} entre os de entrada e o menor entre os de saída em ambos os caso que estiverem entre $[0,1]$;
- **Passo 6:** calcular os pontos de intersecção
- **Passo 7:** traçar o segmento recortado.

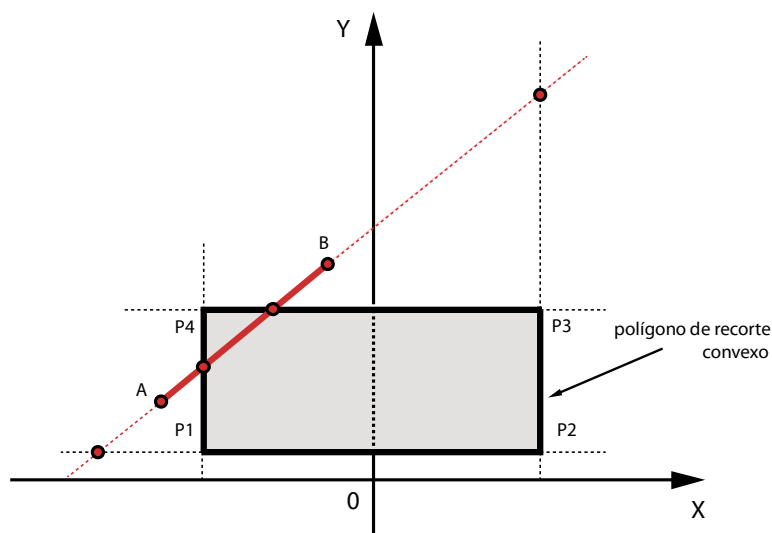


Figura 47: Apenas o ponto de intersecção do segmento AB com a aresta P1P2 possui t_{AB} no intervalo $[0,1]$

Passo 1:

Sabemos que o cálculo de t_{AB} para cada aresta do polígono de recorte é dado po :

$$t_{AB} = \frac{N \cdot (P_A - P_n)}{-N \cdot D}$$

Sendo:

$$\begin{aligned} D &= P_B - P_A \\ D &= (-1 - (-4))\vec{i} + (6 - 2)\vec{j} \\ D &= 3\vec{i} + 4\vec{j} \end{aligned}$$

Cálculo do vetor normal para cada aresta:

$$\begin{aligned} P_1P_2 &= 6\vec{i} + 0\vec{j} & N_{P_1P_2} &= 0\vec{i} - 6\vec{j} \\ P_2P_3 &= 0\vec{i} + 4\vec{j} & N_{P_2P_3} &= 4\vec{i} + 0\vec{j} \\ P_3P_4 &= -6\vec{i} + 0\vec{j} & N_{P_3P_4} &= 0\vec{i} + 6\vec{j} \\ P_4P_1 &= 0\vec{i} - 4\vec{j} & N_{P_4P_1} &= -4\vec{i} + 0\vec{j} \end{aligned}$$

Para cada aresta temos PA-Pn:

$$\begin{aligned} PA - P_1 &= -1\vec{i} + 1\vec{j} \\ PA - P_2 &= -7\vec{i} + 1\vec{j} \\ PA - P_3 &= -7\vec{i} - 3\vec{j} \\ PA - P_4 &= -1\vec{i} - 3\vec{j} \end{aligned}$$

Para cada aresta temos os seguintes vetores normais:

$$\begin{aligned} N_{P_1P_2} \cdot PAP_1 &= -6 \\ N_{P_2P_3} \cdot PAP_2 &= -28 \\ N_{P_3P_4} \cdot PAP_3 &= -18 \\ N_{P_4P_1} \cdot PAP_4 &= 4 \end{aligned}$$

Podemos calcular o t de intersecção com cada aresta:

$$\begin{aligned} t_{P_1P_2} &= -0.25 \\ t_{P_2P_3} &= 2.33 \\ t_{P_3P_4} &= 0.75 \\ t_{P_4P_1} &= 0.33 \end{aligned}$$

Passo 2:

Vamos calcular o produto interno entre as normais e o vetor formado pelo segmento de reta **AB**.

$$N_{P_1P_2} \cdot AB = (0\vec{i} - 6\vec{j}) \cdot (3\vec{i} + 4\vec{j}) = -24 \text{ ponto de saída}$$

$$N_{P_2P_3} \cdot AB = (4\vec{i} + 0\vec{j}) \cdot (3\vec{i} + 4\vec{j}) = 12 \text{ ponto de entrada}$$

$$N_{P_3P_4} \cdot AB = (0\vec{i} + 6\vec{j}) \cdot (3\vec{i} + 4\vec{j}) = 24 \text{ ponto de entrada}$$

$$N_{P_4P_1} \cdot AB = (-4\vec{i} + 0\vec{j}) \cdot (3\vec{i} + 4\vec{j}) = -12 \text{ ponto de saída}$$

Passo 3:

Tabela 1: cálculo do parâmetro t de entrada e saída

Aresta	t para pontos de entrada	t para pontos de saída
P1P2	-0.25	
P2P3		2.33
P3P4		0.75
P4P1	0.33	

Passo 4:

Tabela 2: classificação de pontos de entrada

Aresta	$\text{maior } t \text{ de entrada}$
P4P1	0.33
P1P2	-0.25

Tabela3: classificação de pontos de saída

Aresta	$\text{menor } t \text{ de saída}$
P3P4	0.75
P2P3	2.33

Passo 5:

Temos como maior t de entrada que esteja no intervalo de $[0,1]$ o valor **0.75**. Como menor valor de t de saída no intervalo de $[0,1]$ temos o valor **0.33**. Portanto, chegamos ao intervalo de t **[0,33, 0,75]**.

Passo 6:

A tabela 1 traz os resultados da aplicação do algoritmo de **Cyrus Beck** para este exemplo e dela temos que escolher o maior t_{AB} de entrada que esteja no intervalo de **[0,1]** e o menor

de saída que esteja também no intervalo **[0,1]**. Assim, temos os valores **0,33** e **0,75** que aplicados a equação paramétrica da reta temos o seguinte segmento de reta:

Para $t_{AB} = 0.33$:

$$x_{A'} = x_{iAB} = -4 + (-1 - (-4))0.33 = -3$$

$$y_{A'} = y_{iAB} = 2 + (6 - 2)0.33 = 3.33$$

$$x_{A'} = x_A + (x_B - x_A)t_{AB}$$

$$y_{A'} = y_A + (y_B - y_A)t_{AB}$$

Para $t_{AB} = 0.75$:

$$x_{B'} = x_{iAB} = -4 + (-1 - (-4))0.75 = -1.75$$

$$y_{B'} = y_{iAB} = 2 + (6 - 2)0.75 = 5$$

O que nos leva ao segmento de **A'(-3, 3.33)** a **B'(-1.75, 5)**, figura (48).

Passo 7:

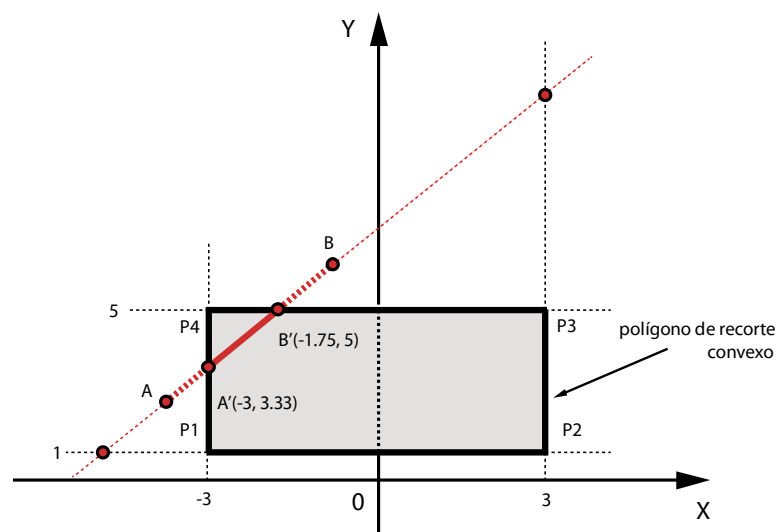


Figura 48: gráfico do resultado do recorte A'B'

7.1.2.3.1 Síntese sobre o Algoritmo de Cyrus-Beck

O algoritmo de **Cyrus Beck** mostra um avanço em relação ao de **Cohen e Sutherland** por evitar cálculos desnecessários relativo a intersecções. Além disso, pode ser usado com regiões de recorte formadas por polígonos convexos, não apenas com polígonos retangulares. Apesar do avanço, não detecta rapidamente os casos triviais. É sempre lembrado na literatura, pois a solução proposta para o problema de recorte de segmentos de retas apresentada mostra uma metodologia interessante.

7.1.2.4 Algoritmo de Liang and Barsky

Este algoritmo é também baseado na equação paramétrica da reta. Tem como especificidade que o recorte é aplicado contra um polígono retangular. Esta característica torna o algoritmo mais eficiente, neste contexto, do que os anteriormente apresentados.

É fácil observar que, no caso de regiões retangulares, os cálculos são simplificados. Neste sentido o algoritmo de You-Dong Liang e Brian A. Barsky procura evitar o cálculo de intersecções que não contribuam para o efetivo recorte.

Leva em consideração os valores do parâmetro t para calcular se um determinado segmento de reta ou parte dele se encontra dentro ou fora da região de recorte (figura 49).

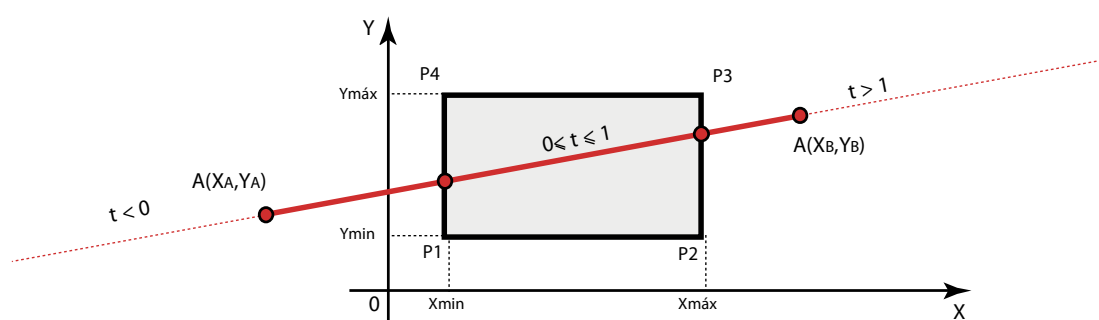


Figura 49: Polígono de recorte retangular

O objetivo do recorte é mostrar apenas os objetos gráficos que estão no interior do polígono de recorte. Este algoritmo destaca a região de recorte, que por sua vez pode ser expressa pelas seguintes inequações:

$$x_{min} \leq x \leq x_{max}$$

$$y_{min} \leq y \leq y_{max}$$

Vimos que a equação paramétrica do segmento de reta AB pode ser traduzida genericamente da seguinte forma:

$$P(t_{AB}) = P_A + (P_B - P_A)t_{AB} \quad \text{com } t_{AB} \text{ entre } [0,1]$$

Especificamente temos:

$$x_{AB} = x_A + (x_B - x_A)t_{AB} = x_A + \Delta_x t_{AB}$$

$$y_{AB} = y_A + (y_B - y_A)t_{AB} = y_A + \Delta_y t_{AB}$$

Podemos escrever:

$$x_{AB} = x_A + \Delta_x t_{AB}$$

$$y_{AB} = y_A + \Delta_y t_{AB}$$

Note que :

$$\Delta_x = x_B - x_A$$

$$\Delta_y = y_B - y_A$$

Colocando a equação paramétrica na definição de recorte anteriormente apresentada, temos:

$$x_{min} \leq x_{AB} \leq x_{max}$$

$$y_{min} \leq y_{AB} \leq y_{max}$$

Substituindo x e y por suas equações paramétricas, temos:

$$x_{min} \leq x_A + \Delta_x t_{AB} \leq x_{max}$$

$$y_{min} \leq y_A + \Delta_y t_{AB} \leq y_{max}$$

Das equações acima, podemos escrever:

$$-\Delta_x t_{AB} \leq x_A - x_{min}$$

$$\Delta_x t_{AB} \leq x_{max} - x_A$$

$$-\Delta_y t_{AB} \leq y_A - y_{min}$$

$$\Delta_y t_{AB} \leq y_{max} - y_A$$

Estas inequações podem ser expressas da seguinte forma:

$$t_{AB} \cdot p_k \leq q_k$$

Com $k=1,2,3$ e 4

$$t_{AB} \leq \frac{q_k}{p_k}$$

$$P_1 = -\Delta_x$$

$$q_1 = x_A - x_{min}$$

$$P_2 = \Delta_x$$

$$q_2 = x_{max} - x_A$$

$$P_3 = -\Delta_y$$

$$q_3 = y_A - y_{min}$$

$$P_4 = \Delta_y$$

$$q_4 = y_{max} - y_A$$

Analisando os parâmetros \mathbf{p} e \mathbf{q} podemos identificar algumas posições do segmento de reta em relação a uma janela de recorte retangular. Para ilustrar esta assertiva vamos considerar que P_1 seja negativo e verificar o significado geométrico desta suposição (figura 50).

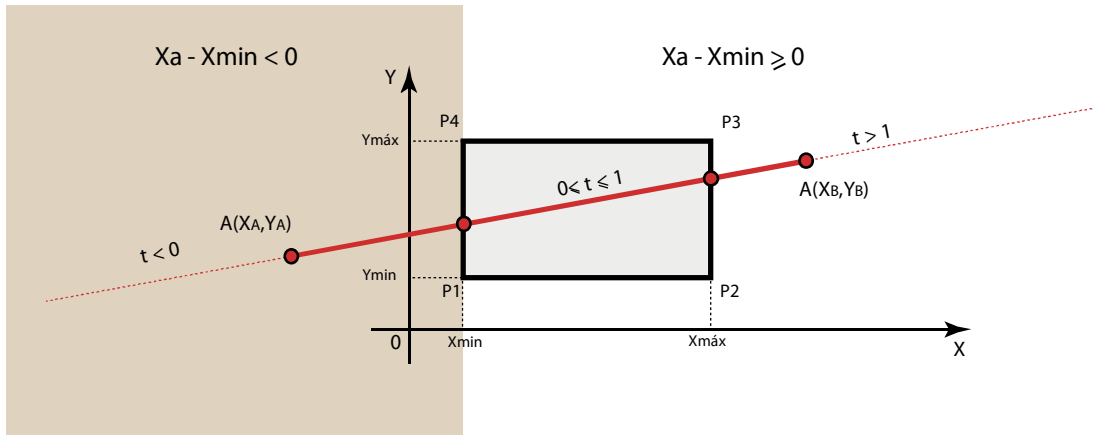


Figura 50: espaço definido $X_A - X_{min} < 0$

Observe que $X_A - X_{min}$ for menor que zero indica que X_{min} é maior que X_A , portanto $q_1 < 0$ refere-se a região a esquerda da janela de recorte. Analogamente a suposição de $q_3 < 0$ temos $y_A - y_{min} < 0$ indica a região inferior a janela de recorte (Figura 51).

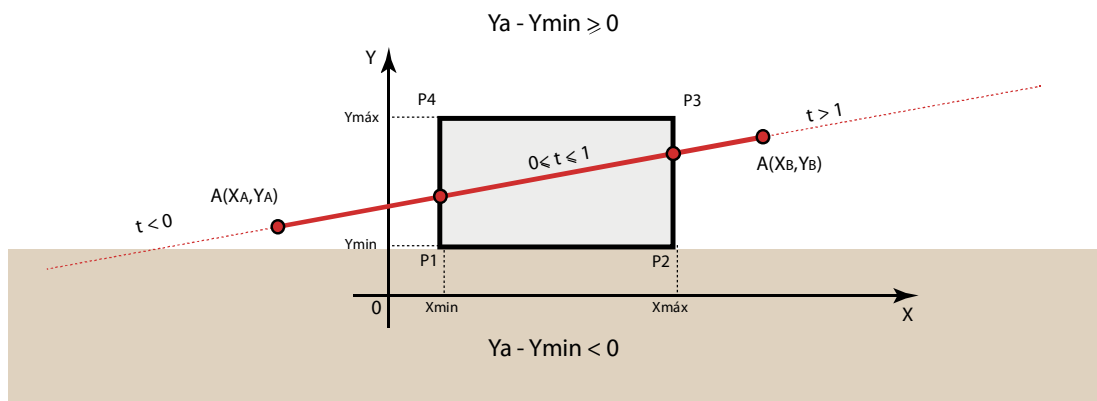


Figura 51: espaço definido $Y_A - Y_{\min} < 0$

Analisando os possíveis valores que as variáveis p_k e q_k possam assumir, podemos resumidamente observar:

- $p_k = 0$ é paralelo a aresta k ;
- $q_k < 0$ o segmento esta completamente fora da janela;
- $q_k \geq 0$ o segmento esta dentro da janela;
- $p_k < 0$ o segmento vem de fora para dentro da janela;
- $p_k > 0$ o segmento vem de dentro para fora da janela.
- $p_k \neq 0$ o valor do parâmetro t pode ser calculado por: $t_{AB} = \frac{q_k}{p_k}$

Algoritmo:

```

Program Liang-Barsky
begin
    dx:= x2-x1; dy:= y2-y1;
    p1:= -dx; q1:= x1-xmin;
    p2:= dx; q2:= xmax-x1;
    p3:= dy; q3:= y1-ymin;
    p4:= dy; q4:= ymax-y1;

    tmin:= 1; tmax:= 0;

    for k:=1 to 4 do
    begin
        if ((p[k]=0) and (q[k]<0)) excluir segmento;
        if (p[k] < 0)
        begin
            t:= q[k]/p[k];
            if (t > tmax) tmax := t;
        end;
        if (p[k] > 0)
    end

```



```

begin
    t:= q[k]/p[k];
    if (t < tmin) tmin:= t;
end;
end;
if (tmax >= tmin) then excluir segmento
    else desenha segmento [tmin,tmax];
end.

```

Exemplo 09: Dada um janela de recorte retangular definida pelos extremos ($x_{min} = 1$, $y_{min} = 1$) e ($x_{max} = 6$, $y_{max} = 5$). Realize o recorte, se necessário, do segmento de reta definido pelos pontos A(-1,2) e B(7,4), utilizando o algoritmo de Liang-Barsky, (figura 52):

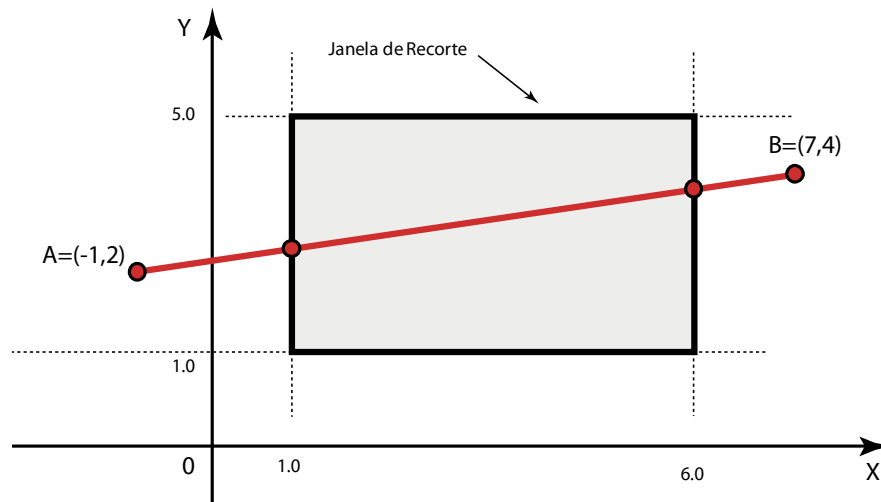


Figura 52: segmento de reta AB cortando aresta esquerda e direita do retângulo de recorte

A solução proposta por **Liang e Barsky** (1999) pode ser resumida nos seguintes passos:

- **Passo 1:** calcular os valores dos parâmetros p_k e q_k ;
- **Passo 2:** testar se $p_k = 0$ para verificar se o segmento AB é paralelo a alguma aresta do polígono de recorte;
- **Passo 3:** testar se $q_k < 0$ se todos os q_k forem simultaneamente negativos o polígono esta fora do polígono de recorte; Caso esta condição se verifique o algoritmo é finalizado. Caso contrário verificar se o segmento esta totalmente dentro da região de recorte ou recorta-lo para atingir esta condição.
- **Passo 4:** inicialize os valores de t com [0,1];
- **Passo 5:** calcule os valores de $\frac{q_k}{p_k}$ para $k = 1,2,3,4$;

- **Passo 6:** Selecione valores de $\frac{q_k}{p_k}$ onde $p_k < 0$ e faça o maior deles como t_1 ;
- **Passo 7:** selecione valores de $\frac{q_k}{p_k}$ onde $p_k > 0$ e faça o menor deles como t_2 ;
- **Passo 8:** se $t_1 < t_2$ calcular os pontos de recortes assim definidos:

Vamos ao **Passo 1:**

$$\begin{aligned}x_A &= -1 & y_A &= 2 \\x_B &= 7 & y_B &= 4\end{aligned}$$

$$\begin{aligned}\Delta_x &= x_B - x_A = 7 - (-1) = 8 \\ \Delta_y &= y_B - y_A = 4 - 2 = 2\end{aligned}$$

$$\begin{aligned}x_{min} &= 1, & y_{min} &= 1 \\ x_{max} &= 6, & y_{max} &= 5\end{aligned}$$

$$\begin{array}{ll}p_1 = -8 & q_1 = -2 \\ p_2 = 8 & q_2 = 7 \\ p_3 = -2 & q_3 = 1 \\ p_4 = 2 & q_4 = 3\end{array}$$

Passo 2:

Como todos os valores de p são diferentes de zero, podemos afirmar que o segmento de reta AB não é paralelo a nenhuma das arestas;

Passo 3:

Analisando os valores de q podemos ver que não são todos negativos, portanto a hipótese de que o segmento está fora da janela de recorte pode ser descartada. Temos agora que prosseguir o algoritmos para determinar os pontos de intersecção com as arestas do retângulo de recorte.

Passo 4:

Vamos inicializar os valores de $t_1 = 0$ e de $t_2 = 1$;

Passo 5:

Cálculo dos valores de t para q_k/p_k com $k=1,2,3,4$

$$t_1 = \frac{q_1}{p_1} = \frac{1}{4}$$

$$t_2 = \frac{q_2}{p_2} = \frac{7}{8}$$

$$t_3 = \frac{q_3}{p_3} = \frac{-1}{2}$$

$$t_4 = \frac{q_4}{p_4} = \frac{3}{2}$$

Passo 6:

Vamos escolher dentre os p_k negativos o maior valor de t .

$$t_1 = 1/4$$

$$t_3 = -1/2$$

Passo 7:

Vamos escolher dentre os p_k negativos o maior valor de t .

$$t_2 = \frac{7}{8}$$

$$t_4 = \frac{3}{2}$$

Passo 8:

Calculo dos pontos de recorte com os valores determinados de t_1 e t_2 .

Para t_1 :

$$x_1 = x_A + t_1 \cdot \Delta_x$$

$$y_1 = y_A + t_1 \cdot \Delta_y$$

$$x_2 = x_A + t_2 \cdot \Delta_x$$

$$y_2 = y_A + t_2 \cdot \Delta_y$$

$$x_1 = -1 + \frac{7}{8} 8 = 6$$

$$y_1 = 2 + \frac{1}{4} 2 = 2.5$$

$$x_2 = -1 + \frac{1}{4} 8 = 1$$

$$y_2 = 2 + \frac{7}{8} 2 = 3.75$$

Graficamente temos:

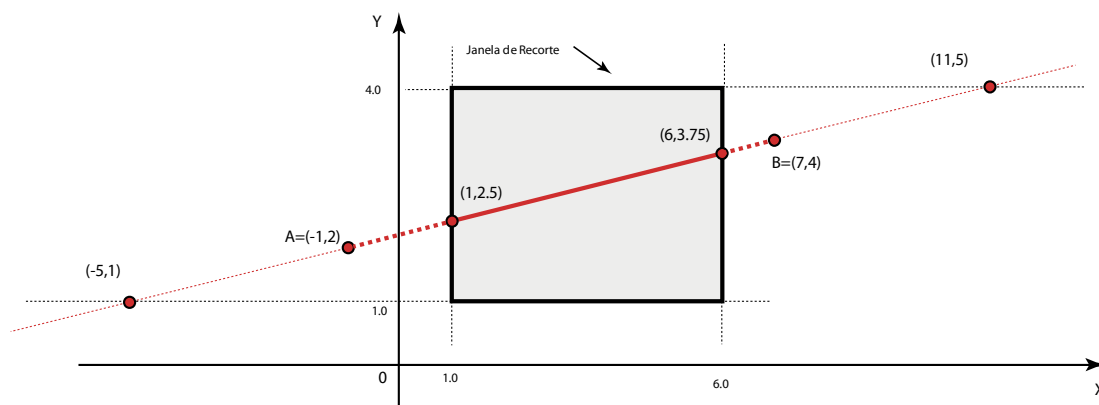


Figura 53: segmento AB recortado

7.1.2.4.1 Síntese sobre o Algoritmo de Liang and Barsky

O algoritmo em de **Liang-Barsky** apresenta ganho de performance em relação aos algoritmos anteriores a ele aqui apresentados. Apesar disso, pode ser aplicado apenas em regiões retangulares de recorte.

7.1.2.5 Algoritmo da Divisão pelo Ponto Médio

Os algoritmos até aqui apresentados para o recorte de segmentos de reta atuam em coordenadas do mundo (**WCS – World Coordinates System**). O Algoritmo de divisão pelo ponto médio pode atuar em coordenadas do dispositivo, ou seja em um meio discreto. É fundamental do ponto de vista histórico e pedagógico a sua apresentação. Ele é utilizado

em conjunto com o algoritmo de **Cohen e Sutherland**. Neste sentido, ele é aplicado para a determinação do ponto de intersecção, uma vez que, a identificação da situação do seguimento de reta quanto a possibilidade de ser recortado é feita pelo algoritmo de **Cohen e Sutherland**.

O algoritmo em tela, consiste em dividir o segmento de reta em seu ponto médio gerando dois segmentos (figura 54). Em seguida, avaliá-los utilizando os critérios de **Cohen-Sutherland** para verificar a situação de visibilidade do segmento em relação ao polígono de recorte.

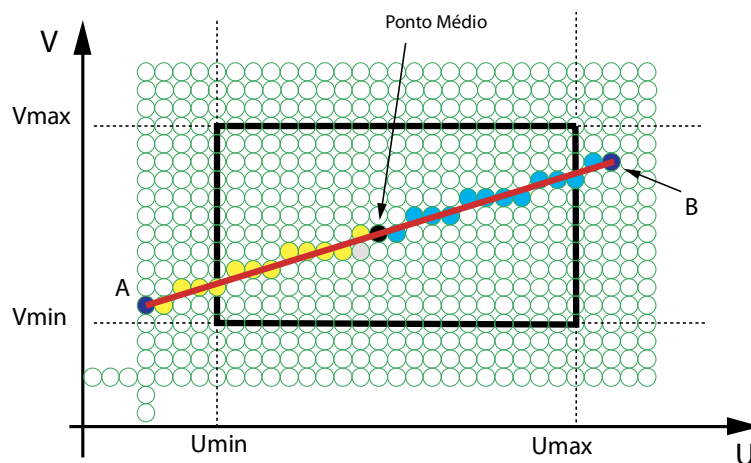


Figura 54: segmentos de reta gerados pela divisão de AB

De acordo com **Cohen e Sutherland**, os seguimentos podem ser detectados como **visíveis**, **não visíveis** e **sujeitos ao recorte**. O algoritmo da divisão pelo ponto médio aplica iterativamente a divisão do segmento cuja condição é detectada como **sujeito ao recorte** até que os segmentos resultantes deste procedimento atinjam um determinado critério de terminalidade, por exemplo, a chegada à condição de **visível** ou **não visível**. Além deste critério, pode ser especificado um limite de tolerância que indique quanto o ponto médio calculado está próximo do valor que representa a aresta do polígono de recorte.

O ponto médio do segmento **AB**, aqui denominado de P_{MAB} , pode ser calculado da seguinte forma:

$$P_{MAB} = \frac{P_A + P_B}{2}$$

$$P_{MAB} = (x_{MAB}, y_{MAB})$$

$$x_{MAB} = \frac{x_A + x_B}{2}$$

$$y_{MAB} = \frac{y_A - y_B}{2}$$

De forma pragmática podemos expressar o método por meio do seguinte algoritmo:

Algoritmo da Divisão pelo Ponto Médio

- Passo 1:** Leitura das das coordendas dos pontos **A** e **B** via **viewport** ou **janela**;
- Passo 2:** leitura dos extremos do polígono de recorte (**janela** ou **viewport**);
- Passo 3:** determinação dos **outcodes** dos extremos do segmento AB (inicializar **outcodes** com **0000**);
- Passo 4:** checagem da visibilidade do segmento de reta por meio do algoritmo de **Cohen-Sutherland**;
- Passo 5:** dividir pela metade os segmentos detectados como parcialmente visíveis e repetir os passos **3** a **5** até que os **outcodes** apontem segmentos como **visíveis** ou **não visíveis**.ou seja satisfeito um determinado critério de terminalidade

Exemplo 10: Vamos calcular as intersecções de um segmento de reta definido pelos pontos **A(4,7)** e **B(30,15)** com a **viewport** com coordenadas (**umin=8, vmin=6**) e (**umáx=28, vmáx=17**), (figura 55), utilizando lo método da divisão pelo ponto médio.

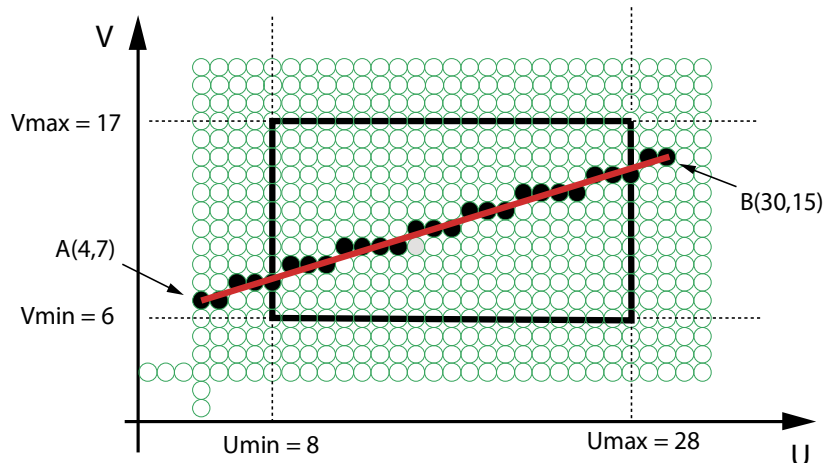


Figura 55: segmentos de reta gerados pela divisão de AB

Inicialmente vamos calcular os **outcodes** dos extremos do segmento de reta **AB** para verificar se o segmento está totalmente dentro da **viewport**.

Ponto A: **0001**

Ponto B: **0010**

Como ambos os **outcodes** não são **0000**, podemos afirmar que o segmento não está totalmente contido na **viewport**. Vamos em seguida, calcular o **AND lógico** para verificar se o seguimento está totalmente fora da região de recorte ou se é necessário o recorte.

AND Lógico entre **A** e **B**: **0000**

O resultado do **AND lógico** indica que o segmento não está totalmente fora do **viewport**, portanto, deve ser recortado. Desta forma, temos que fazer com que ambos os **outcodes** sejam **0000** concomitantemente. Vamos iniciar o procedimento com o ponto **A** cujo **outcode** indica uma intersecção com a aresta esquerda da **viewport**. O ponto de intersecção da reta com a referida aresta possui as seguintes coordenadas:

$$I_{ae} = (u_{min}, v_{iae})$$

Onde:

I_{ae} : Intersecção com a aresta esquerda

u_{min} : abscissa do limite mínimo da janela

v_{iae} : Ordenada da intersecção com a aresta esquerda

Como critério de terminalidade, neste exemplo específico, podemos utilizar um limite de tolerância entre os valores da abscissa do ponto médio calculado (PM_i) e a da aresta (u_{min}). Neste exemplo arbitraremos 0,1. Assim, o critério de terminalidade fica:

$$|PM_i - u_{min}| < 0,1$$

Ou seja:

$$|PM_i - 8| < 0,1$$

Vamos calcular o ponto médio do segmento de reta **AB**. Chamaremos este ponto de **Ponto Médio 1**, assim expresso **PM1**, figura (56):

Algebricamente, temos:

$$P_{M1} = (x_{M1}, y_{M1})$$

$$x_{M1} = \frac{x_A + x_B}{2} = \frac{30 + 4}{2} = 17$$

$$y_{M1} = \frac{y_A + y_B}{2} = \frac{15 + 7}{2} = 11$$

$$P_{M1} = (17, 11)$$

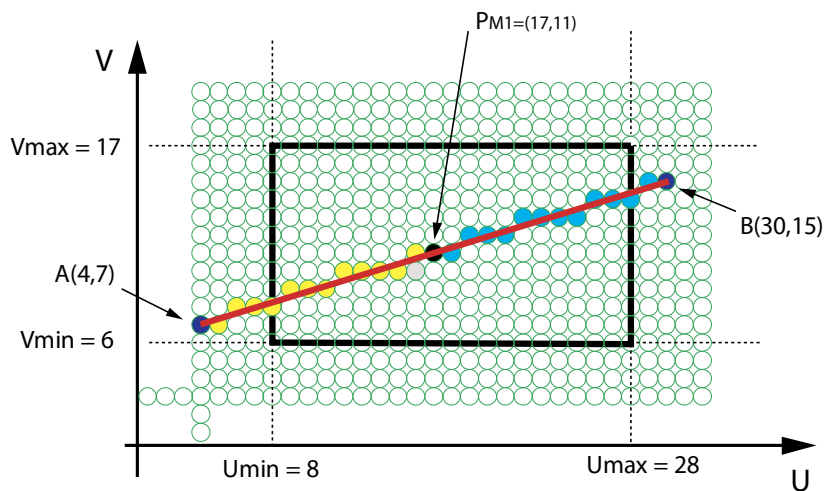


Figura 56: segmentos de reta gerados pela divisão de AB

Vamos testar o critério de terminalidade:

$$|PM_i - 8| < 0,1$$

$$|17 - 8| < 0,1$$

$$|9| < 0,1$$

Observamos que 9 não é menor que 0.1. Assim, continuamos o processo iterativo. Ou seja, temos agora dois segmentos para serem avaliados. O segmento **A-PM1** e o segmento **PM1-B**. Vamos avaliar a condição dos segmentos pelo método de **Cohen e Sutherland**. Inicialmente iremos analisar apenas o trecho **A-PM1** do segmento inicial e, após, avaliaremos o trecho **PM1-B**. Para isto, vamos determinar os **outcodes** dos extremos deste segmento.

Ponto **A**: **0001**

Ponto **PM1**: **0000**

Como ambos os **outcodes**, simultaneamente, não são **0000**, podemos afirmar que o segmento não está totalmente inserido na **viewport**. Dando prosseguimento, vamos verificar se ele se encontra totalmente fora do **viewport** fazendo o **AND lógico** do ponto **A** com o ponto **PM1**.

AND Lógico entre **A** e **PM1**: **0000**

O **AND lógico** indica que temos que continuar a busca pelo ponto de intersecção para executar o recorte contra a aresta esquerda da **viewport**.

Vamos calcular o ponto médio do segmento de reta **A-PM1**. Chamaremos este ponto de **Ponto Médio 2**, assim expresso **PM2**:

$$P_{M2} = (x_{M2}, y_{M2})$$

$$x_{M2} = \frac{x_A + x_{PM}}{2} = \frac{4 + 17}{2} = 10.5$$

$$y_{M1} = \frac{y_A + y_{PM}}{2} = \frac{7 + 11}{2} = 9$$

$$P_{M2} = (10.5, 9)$$

Teste do critério de terminalidade:

$$|PM_i - 8| < 0,1$$

$$|10.5 - 8| < 0,1$$

$$|2.5| < 0,1$$

Como o critério de terminalidade não foi satisfeito, continuamos o processo iterativo. Vamos dividir o próximo seguimento.

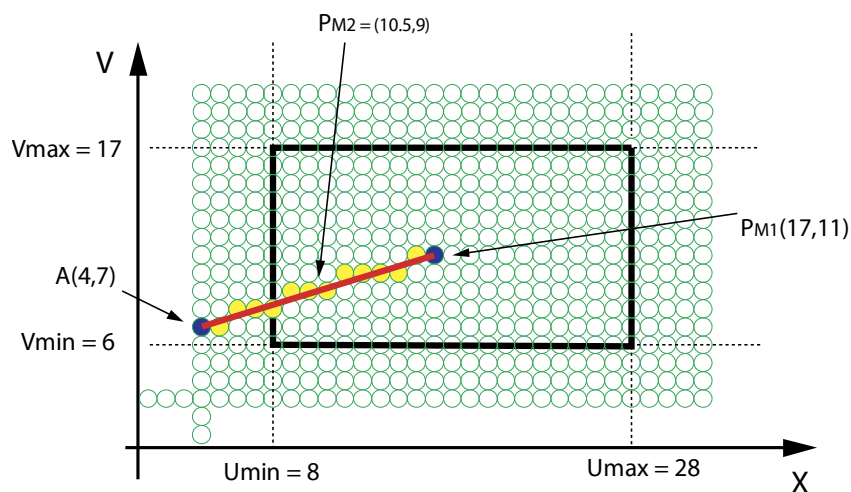


Figura 57: segmentos de reta gerados pela divisão de AB

Temos novamente dois segmentos para serem avaliados. O segmento **A-PM2** e o segmento **PM2-PM1** (figura 57). Vamos determinar os **outcodes** dos extremos destes segmentos.

Segmento **A-PM2**:

Ponto **A**: **0001**

Ponto **PM2**: **0000**

Segmento **PM1-PM2**:

Ponto **PM1**: **0000**

Ponto **PM2**: **0000**

No segmento **A-PM2** ambos os **outcodes**, simultaneamente, não são **0000**, podemos afirmar que o segmento não está totalmente inserido na **viewport**. Temos que verificar se

ele se encontra totalmente fora do **viewport** fazendo o **AND lógico** do ponto **A** com o ponto **PM2**.

AND Lógico entre **A** e **PM2**: **0000**

O **AND lógico** indica que devemos prosseguir dividindo o segmento ao meio para encontrar o ponto de intersecção.

No caso do segmento **PM2-PM1**, ambos os **outcodes** são **0000**, o que indica que este trecho do segmento de reta está totalmente dentro da **viewport**. Desta forma, descartamos este trecho do segmento na nossa busca pela intersecção.

Vamos agora calcular o ponto médio do segmento de reta **A-PM2**. Chamaremos este ponto de **Ponto Médio 3**, assim expresso **PM3**:

$$P_{M3} = (x_{M3}, y_{M3})$$

$$x_{M3} = \frac{x_A + x_{PM}}{2} = \frac{4 + 10.5}{2} = 7.25$$

$$y_{M3} = \frac{y_A + y_{PM}}{2} = \frac{7 + 9}{2} = 8$$

$$P_{M3} = (7.25, 8)$$

Teste do critério de terminalidade:

$$|PM_i - 8| < 0,1$$

$$|7.25 - 8| < 0,1$$

$$|0.75| < 0,1$$

Novamente o critério de terminalidade não foi satisfeito, continuamos o processo iterativo. Vamos dividir o próximo seguimento.

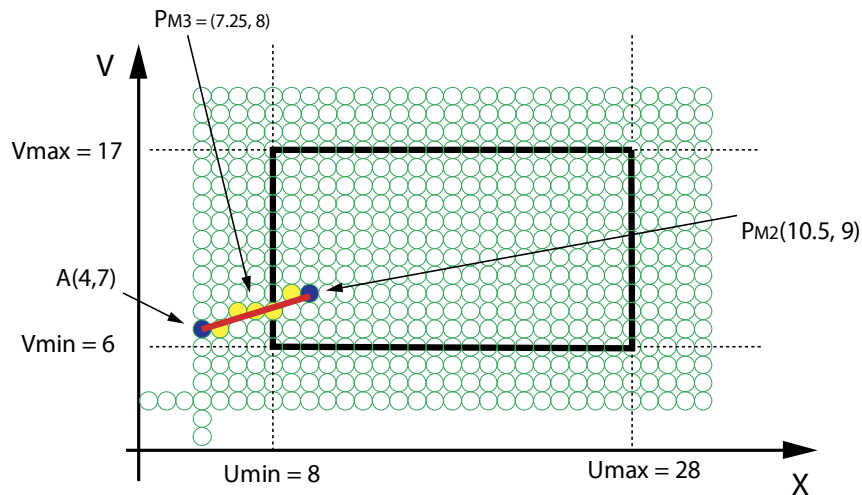


Figura 58: segmentos de reta gerados pela divisão de AB

Mais uma vez temos dois segmentos para serem avaliados. O segmento **A-PM3** e o segmento **PM3-PM2**, (figura 58). Vamos determinar os **outcodes** dos extremos destes segmentos.

Segmento **A-PM3**:

Ponto **A**: **0001**

Ponto **PM3**: **0001**

Segmento **PM3-PM2**:

Ponto **PM3**: **0001**

Ponto **PM2**: **0000**

No segmento **A-PM3** ambos os **outcodes**, simultaneamente, não são **0000**, podemos afirmar que o segmento não está totalmente inserido na **viewport**. Temos que verificar se ele se encontra totalmente fora do **viewport** fazendo o **AND lógico** do ponto **A** com o ponto **PM3**.

AND Lógico entre A e PM3: 0001

O **AND lógico** indica que o segmento **A-PM3** esta totalmente fora da viewport, portando, deve ser descartado.

No caso do segmento **PM3-PM2**, os **outcodes**, simultaneamente, não são **0000**, o que indica que este trecho do segmento de reta deve subdividido em busca do ponto de intersecção.

Vamos agora calcular o ponto médio do segmento de reta **PM3-PM2**. Chamaremos este ponto de **Ponto Médio 4**, assim expresso **PM4**:

$$P_{M4} = (x_{M4}, y_{M4})$$

$$x_{M4} = \frac{x_{PM3} + x_{PM}}{2} = \frac{7.25 + 10.5}{2} = 8.875$$

$$y_{M4} = \frac{y_{PM3} + y_{PM}}{2} = \frac{8 + 9}{2} = 8.5$$

$$P_{M4} = (8.875, 8.5)$$

Teste do critério de terminalidade:

$$|PMi - 8| < 0,1$$

$$|8.875 - 8| < 0,1$$

$$|0.875| < 0,1$$

Mais uma vez o critério de terminalidade não foi satisfeito, continuamos o processo iterativo. Vamos dividir o próximo seguimento.

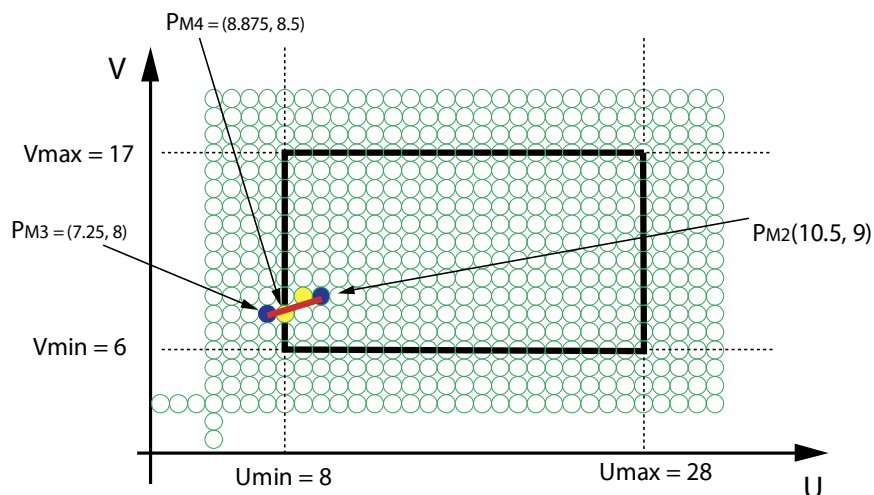


Figura 59: segmentos de reta gerados pela divisão de AB

Analogamente a situações anteriores, temos dois segmentos para serem avaliados. O segmento **PM3-PM4** e o segmento **PM4-PM2**, (figura 59). Vamos determinar os **outcodes** dos extremos destes segmentos.

Segmento **PM3-PM4**:

Ponto A: **0001**

Ponto PM3: **0000**

Segmento **PM4-PM2**:

Ponto **PM3**: **0000**

Ponto **PM4**: **0000**

No segmento **PM3-PM4**, os **outcodes**, simultaneamente, não são **0000**, podemos afirmar que o segmento não está totalmente inserido na **viewport**. Temos que verificar se ele se encontra totalmente fora do **viewport** fazendo o **AND lógico** do ponto **PM3** com o ponto **PM4**.

AND Lógico: **0000**

O **AND lógico** indica que o segmento **PM3-PM4** não está totalmente fora da viewport, portando, deve ser recortado e subdividido em busca do ponto de intersecção.

No caso do segmento **PM4-PM2**, os **outcodes**, simultaneamente, são **0000**, o que indica que este trecho do segmento de reta deve descartado

Vamos agora calcular o ponto médio do segmento de reta **PM3-PM4**. Chamaremos este ponto de **Ponto Médio 5**, assim expresso **PM5**:

$$P_{M5} = (x_{M5}, y_{M5})$$

$$x_{M5} = \frac{x_{PM3} + x_{PM4}}{2} = \frac{7.25 + 8.875}{2} = 8.06$$

$$y_{M5} = \frac{y_{PM3} + y_{PM4}}{2} = \frac{8 + 8.5}{2} = 8.25$$

$$P_{M5} = (8.06, 8.25)$$

Teste do critério de terminalidade:

$$|PMi - 8| < 0,1$$

$$|8.06 - 8| < 0,1$$

$$|0.06| < 0,1$$

Neste caso o critério de terminalidade é satisfeito. Desta forma, podemos afirmar que o ponto de intersecção com a aresta esquerda do retângulo de recorte é o ponto:

$$I_{ae} = (8.06, 8.25)$$

Com truncamento:

$$I_{ae} = (8, 8)$$

É Fácil observar que **PM4** e **PM5** são o mesmo ponto, portanto seus outcodes são iguais (**0001**) e mostram que estão fora da janela de recorte. Assim, fica determinado o ponto de intersecção o ponto **(8,8)**.

Realizando cálculos análogos aos feitos com o trecho **A-PM1** com o trecho **PM1-B** do seguimento de reta **AB**, encontraremos como ponto de intersecção com a aresta direita do polígono de recorte o ponto **(28,14)**. A figura 60, ilustra graficamente os resultados das iterações.

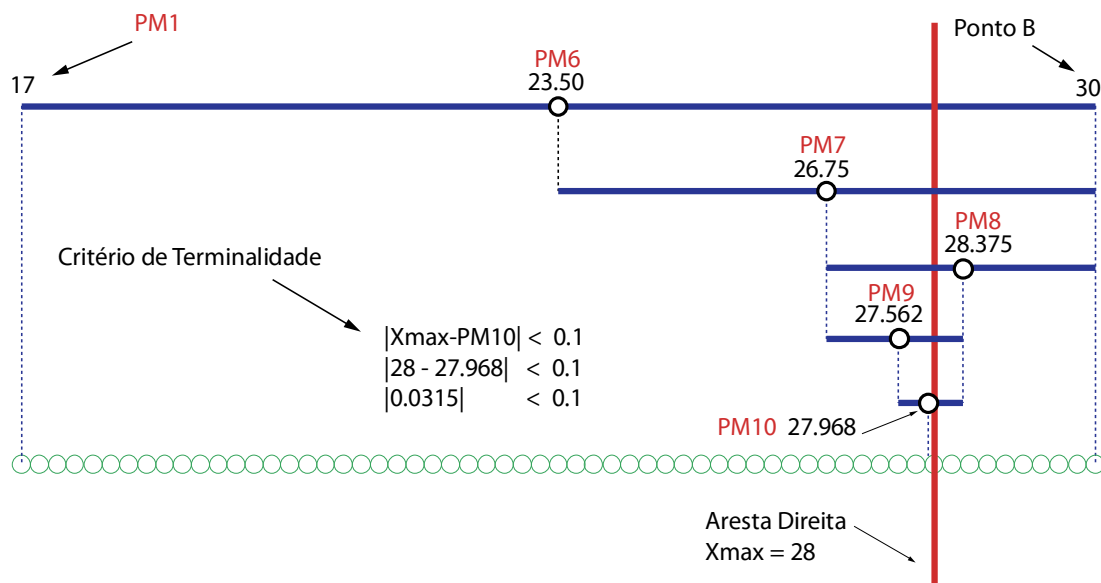


Figura 60: segmentos de reta gerados pela divisão do trecho **PM1-B**

Com o cálculo da intersecção do segmento de reta **AB** com as arestas esquerda e direita do retângulo de recorte podemos desenhar o segmento **AB** recortado (figura 61).

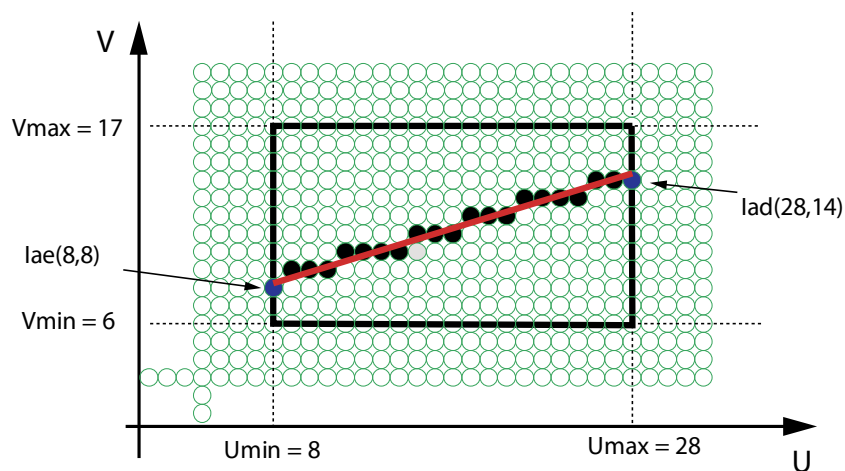


Figura 61: segmentos de reta gerados pela divisão de AB

7.1.2.5.1 Síntese sobre o Algoritmo da Divisão pelo Ponto Médio

Este algoritmo pode ser utilizado no espaço do dispositivo (**DC**) ou no espaço do mundo (**WCS**). Embora exija muitas iterações, possui contribuições históricas e pedagógicas para o estudo do recorte de segmentos de reta. No **exemplo 10**, propositalmente apresentamos graficamente os pixels do dispositivo para mostrar que o número de subdivisões realizadas para encontrar o ponto de intersecção é sempre um múltiplo de 2, ou seja 2^n , onde n indica o número de subdivisões. Desta forma, tomando o número de pixels do dispositivo gráfico como NPD (número pixels do dispositivo), n pode ser assim calculado:

$$n = \log_2 NPD$$

No exemplo anterior, tínhamos um dispositivo com resolução de 32 x 28 pixels. Aplicando a equação acima:

$$n = \log_2 32 = 5$$

Vale observar que, neste exemplo, com no máximo cinco iterações, aplicando o algoritmo da divisão pelo ponto médio, chegaríamos ao ponto de intersecção.

7.1.2.6 Algoritmo da Nicholl-Lee-Nicholl

O algoritmo proposto por **Nicholl-Lee-Nicholl** tenta evitar cálculos desnecessários e determinar com mais precisão do que os algoritmos aqui apresentados as intersecções inevitáveis com as arestas do polígono de recorte. Ele, analogamente ao algoritmos de **Cohen-Sutherland**, divide o espaço de trabalho com a janela de recorte em nove regiões (figura 71).

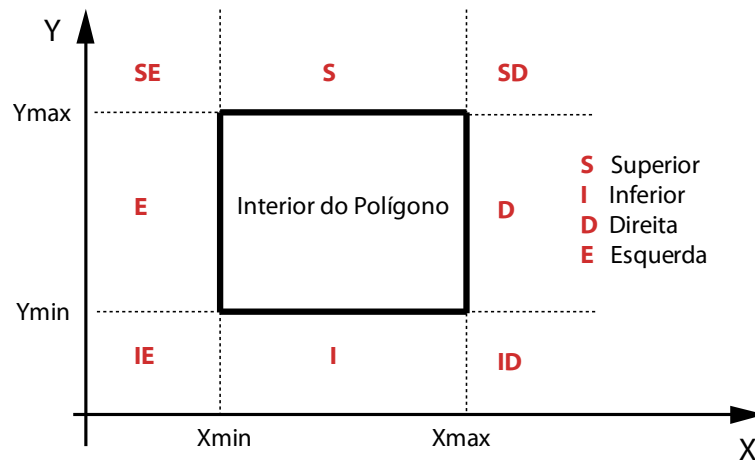


Figura 71: regiões propostas por Nicholl-Lee-Nicholl

Esta proposta considera possíveis localizações de um segmento de reta **AB** a ser recortado por um polígono retangular de recorte. Para desenvolvimento do algoritmo esta solução fixa um dos pontos do segmento de reta para calcular a possível localização do outro e, dependendo de onde estes pontos se encontram, calcular as eventuais intersecções.

Para mostrar o funcionamento do algoritmo vamos considerar um segmento de reta **AB** e um retângulo de recorte. Arbitrariamente o algoritmo fixa um dos pontos do segmento de reta, no caso o ponto **A**, em regiões pré-determinadas e busca a localização do outro ponto (**B**). **Sutherland e Hodgman** mostraram três situações predominantes para fixação do ponto **A** que representam as demais possíveis. A ideia de fixar um ponto e encontrar a posição do outro afim de determinar as eventuais intersecções com as arestas do polígono de recorte facilita a identificação da necessidade de recorte dando eficiência ao algoritmo, Figura 72.

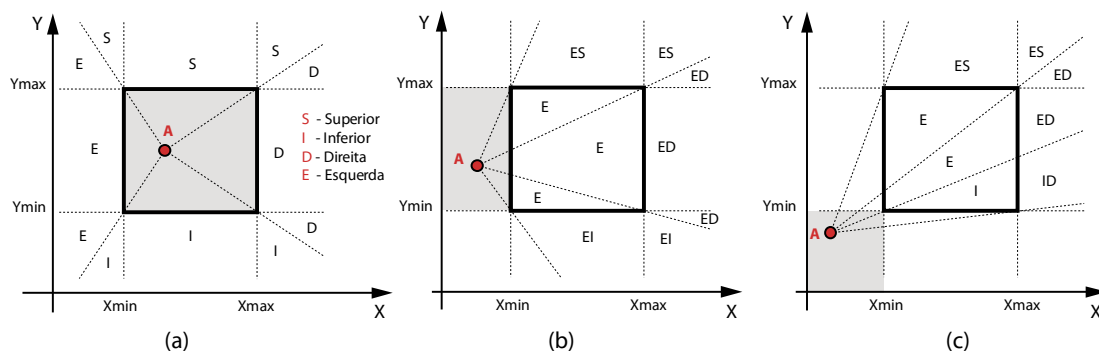


Figura 72: posições arbitrárias do ponto e possíveis regiões de recorte

Temos desta forma, três pontos de partidas para o ponto **A** do segmento de reta e dele projetamos retas que cruzam os vértices do retângulo de recorte. Um suponto que esteja no interior da janela de recorte (Tipo a, figura 72 (a)). Um que esteja em uma das regiões superior, inferior, direita ou esquerda externas a janela de recorte (tipo b, figura 72(b)). E por último um que esteja em regiões que representam os cantos externos a janela de recorte (tipo c, figura 72(c)). Com efeito, essa técnica permite identificar as arestas do retângulo de recorte com que o segmento **AB** irá se intersectar, evitando assim, o cálculo de intersecções desnecessários. Vamos ilustra a aplicação do algoritmo com um exemplo.

Exemplo 11:

Considere a figura 73. Vamos calcular a intersecção do segmento de reta **AB** com o retângulo de recorte utilizando o algoritmo de Nicholl-Lee-Nicholl.

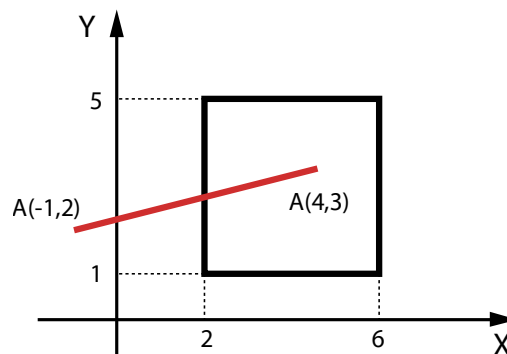


Figura 73: segmento de reta **AB** a ser recortado

Analogamente a outros exemplos vamos relizar a sua solução deste problema discretizando seus passos.

Passo 1: vamos determinar onde está situado o ponto **A** e com isso determinar o tipo de enquadramento, figura 72.

A região em que se encontra o ponto **A** pode ser determinada utilizando-se o método de **Cohen-Sutherland**. Neste caso, temos o seguinte **outcode**: **0001** que indica que o ponto esta na região estritamente a esquerda da janela de recorte. Portanto, temos um enquadramento do **tipo b**.

Agora, temos que descobrir onde se encontra o ponto B. Inicialmente vamos verificar se ele se encontra em uma região a esquerda da janela de recorte. Isto pode ser feito testando-se se $x_B < x_{min}$, figura 74.

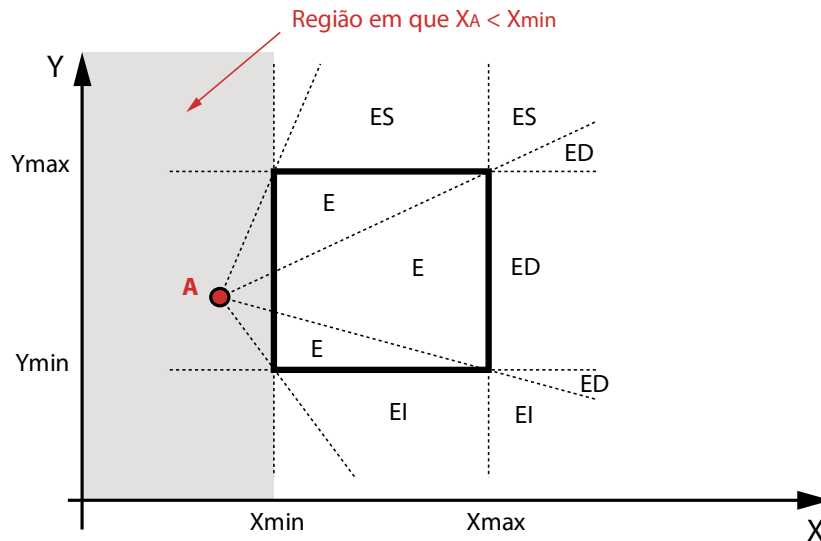


Figura 74: segmento de reta **AB** a ser recortado

Temos $x_{min} = 2$ e $x_B = 4$ ou seja x_B não é menor que x_{min} . Desta forma, o ponto **B** não se encontra nesta região. Temos que continuar nossa busca em outras regiões. Neste sentido, vamos verificar se o ponto **B** se encontra a esquerda ou a direita da reta formada pelo ponto **A** e o vértice superior esquerdo do retângulo de recorte, figura 75.

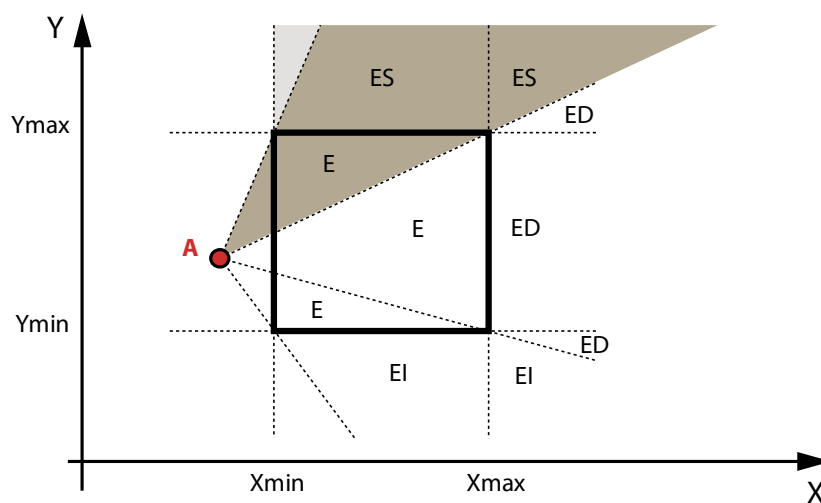


Figura 75: regiões a esquerda e a direita da reta formada pelo ponto **A** e o vértice superior esquerdo já janela

A determinação da posição de **B** em relação em que lado está da reta formada pelo ponto A e o vértice superior esquerdo da janela de recorte pode ser feita avaliando-se o sinal do vetor resultante do produto vetorial entre os vetores AB e AVse (se: superior esquerdo).

De forma geral podemos determinar se um ponto $P(x, y)$ está localizado a esquerda ou a direita de um segmento de reta definido pelo pontos $A(x_A, y_A)$ e $B(x_B, y_B)$, valendo-se do produto vetorial entre os vetores **AB** e **AP**. Ocorre que o produto vetorial entre dois vetores produz um vetor perpendicular ao plano formado pelo dois primeiros. Se o ponto P estiver a direita o vetor resultante do produto vetorial apontará para $-\vec{k}$, figura 76.

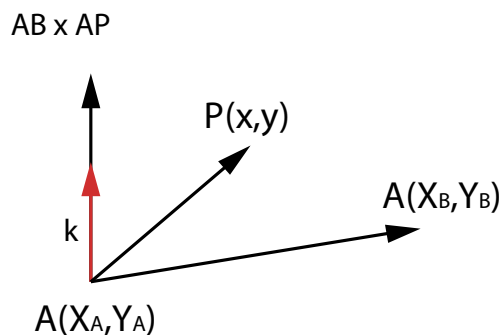


Figura 76: direção de K em relação ao produto vetorial

Podemos escrever os vetores

$$AB = (x_B - x_A)\vec{i} + (y_B - y_A)\vec{j}$$

$$AP = (x_P - x_A)\vec{i} + (y_P - y_A)\vec{j}$$

$$AB \times AP = \begin{bmatrix} i & j & k \\ (x_B - x_A) & (y_B - y_A) & 0 \\ (x_P - x_A) & (y_P - y_A) & 0 \end{bmatrix}$$

$$AB \times AP = ((y_B - y_A) \cdot 0 - (y_P - y_A) \cdot 0)\vec{i} - ((x_B - x_A) \cdot 0 - (x_P - x_A) \cdot 0)\vec{j} + ((x_B - x_A) \cdot (y_P - y_A) - (x_P - x_A) \cdot (y_B - y_A))\vec{k}$$

$$AB \times AP = (x_{Vse} - x_A) \cdot (y_P - y_A) - (x_P - x_A) \cdot (y_{Vse} - y_A)\vec{k}$$

Ou seja, basta avaliar o sinal da expressão acima fazendo o produto vetorial de AB x AVse:

$$(x_{Vse} - x_A) \cdot (y_P - y_A) - (y_{Vse} - y_A) \cdot (x_P - x_A)$$

$$(2 - (-1)).(3 - 2) - (5 - 2).(4 - (-1)) = -12$$

Como o valor é negativo o ponto está a direita do segmento. Assim, vamos continuar o processo e verificar se o ponto está a esquerda ou a direita da reta formada pelo ponto A e o vértice superior direito do retângulo de recorte (figura 77).

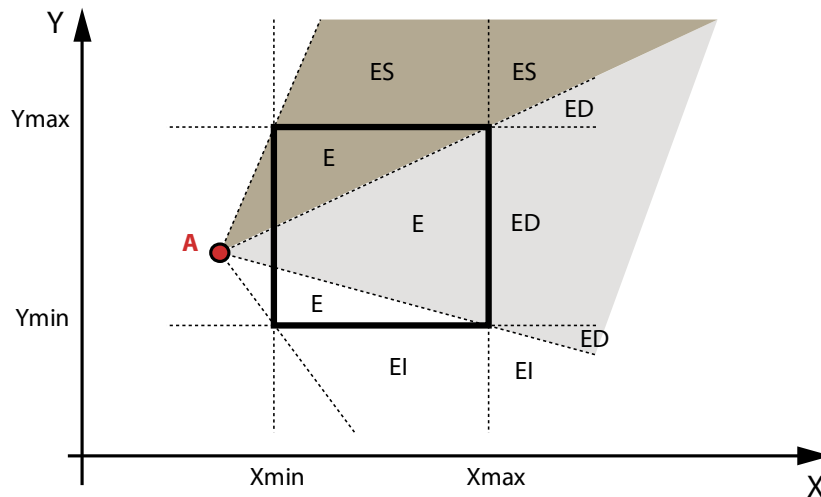


Figura 77: direção de K em relação ao produto vetorial

Para saber se o ponto B está a direita ou a esquerda da reta definida pelo ponto A e o vértice superior direito do retângulo de recorte vamos mais uma vez recorrer ao produto vetorial de AB com AVsd (sd: superior direito).

$$(x_{Vsd} - x_A).(y_P - y_A) - (y_{Vs} - y_A).(x_P - x_A)$$

$$(6 - (-1)).(3 - 2) - (5 - 2).(4 - (0 - (-1))) = -2$$

Como o valor é negativo o ponto está a direita do segmento. Assim, vamos continuar o processo e verificar se o ponto está a esquerda ou a direita da reta formada pelo ponto A e o vértice inferior direito do retângulo de recorte (figura 78).

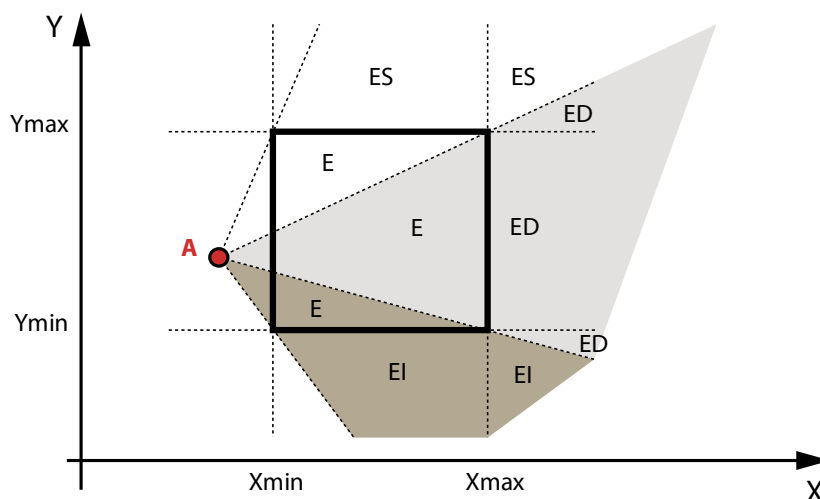


Figura 78: direção de K em relação ao produto vetorial

Para saber se o ponto B está à direita ou à esquerda da reta definida pelo ponto A e o vértice inferior direito do retângulo de recorte vamos mais uma vez recorrer ao produto vetorial de AB com AV_{id} (id: inferior direito).

$$(x_{Vid} - x_A) \cdot (y_P - y_A) - (y_{Vid} - y_A) \cdot (x_P - x_A)$$

$$(6 - (-1)) \cdot (3 - 2) - (5 - 2) \cdot (2 - (0 - (-1))) = 4$$

Como o valor é positivo o ponto está à esquerda do segmento. Assim, sabemos que o segmento de reta AB corta a aresta esquerda do retângulo de recorte. Portanto, podemos calcular a interseção com esta aresta.

Aresta esquerda (ae):

$$x_{ae} = x_{min} + (x_{min} - x_{min})t_{ae} = x_{min}$$

$$y_{ae} = y_{min} + (y_{máx} - y_{mín})t_{ae}$$

$$x_{ae} = x_{min}$$

$$y_{ae} = y_{min} + (y_{máx} - y_{mín})t_{ae}$$

$$x_{ae} = 2$$

$$y_{ae} = 1 + (5 - 1)t_{ae}$$

$$x_{AB} = x_A + (x_B - x_A)t_{AB}$$

$$y_{AB} = y_A + (y_B - y_A)t_{AB}$$

$$x_{AB} = -1 + (4 - (-1))t_{AB}$$

$$y_{AB} = 2 + (3 - 2)t_{AB}$$

Igualando $x_{ae} = x_{AB}$, temos:

$$2 = -1 + (4 - (-1))t_{AB}$$

$$t_{AB} = 3/5$$

$$x_{AB} = 2$$

$$y_{AB} = 2.6$$

O segmento é recortado ficando:

$$A' = (-4, 2.6)$$

$$B = (4, 3)$$

Avaliando os **outcodes** destes pontos temos:

A': 0000

B: 0000

graficamente

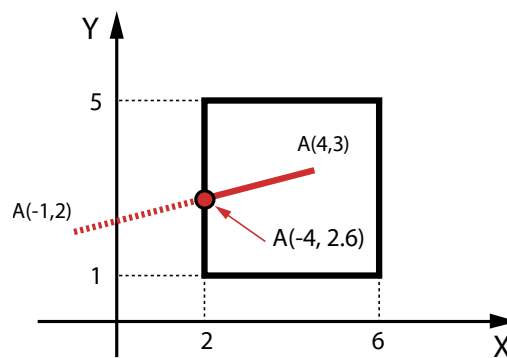


Figura 79: segmento de reta **AB** a ser recortado

7.1.2.6.1 Síntese sobre o Algoritmo da Nicholl-Lee-Nicholl

7.1.3 Recorte de Polígonos

Inicialmente a apresentação de alguns algoritmos que tratam do recorte de polígonos é preciso lembrar que polígonos são figuras planas fechadas limitadas por segmentos de retas que intesectam outros dois extremos (figura 80). Os segmentos de retas que formam o polígono são chamados de lados dos polígono.



Figura 80: exemplos de polígonos

Da definição acima podemos mostrar algumas figuras planas que não formam polígonos por não atenderem tais especificações. A figura 81, ilustra alguns casos. O primeiro caso (a) as retas não se fecham e algumas não ligam dois extremos. No caso (b) os extremos de uma reta ligam apenas a outro lado e não a dois lados. No caso (c) a figura, embora plana, não é formada apenas por segmentos de retas.

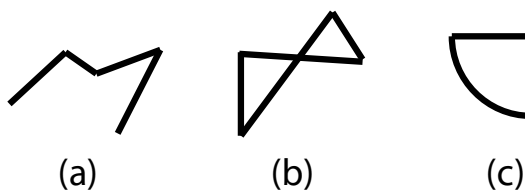


Figura 81: exemplos que formam polígonos

Os polígonos podem ser classificados em côncavos e convexos. Dentre os convexos em regulares e irregulares. São chamados de convexos (figura 82a) quando uma reta definida por dois pontos pertencentes ao interior do polígono não cortam seus lados. Caso corte, o polígono é côncavo (figura 82b).

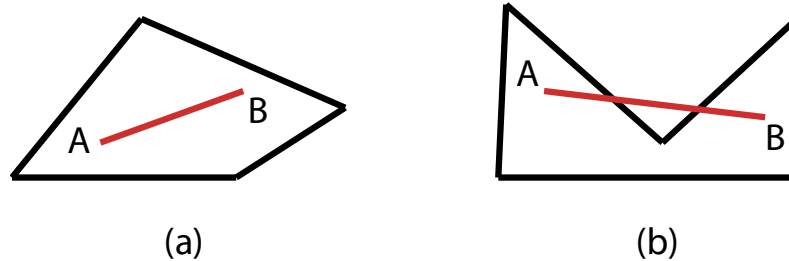


Figura 82: exemplos que formam polígonos

Os polígonos são chamados regulares quando possuem lados de mesmo comprimento e ângulos de mesma amplitude (figura 83).

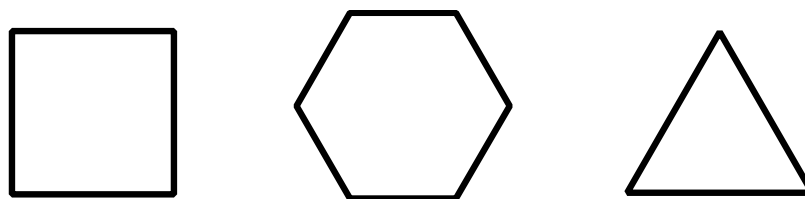


Figura 83: exemplos que formam polígonos

7.1.3.1 O Método de Sutherland e Hodgman

O método de **Sutherland e Hodgman** para recorte de polígonos enumera os vértices do polígono a ser recortado em uma determinada sequência obedecendo um sentido arbitrário. O método percorre os lados do polígono a ser recortado e os avalia para classificá-los em relação à aresta da janela de recorte (figura 84).

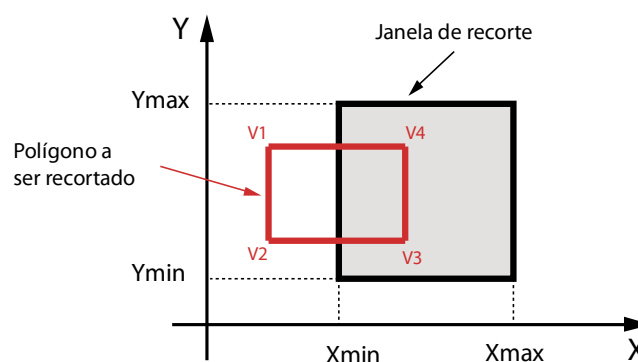


Figura 84: exemplos que formam polígonos

A avaliação leva em conta quatro situações predeterminadas dos lados do **polígono** a ser recortado em relação à **janela de recorte** que são tipificadas em:

- (a) lado do polígono totalmente fora da janela de recorte;
- (b) lado **entrante**, possui um vértice inicial (VI) fora da janela e final (VF) dentro dela;
- (c) lado do polígono totalmente dentro da janela de recorte;
- (d) lado **sainte**, possui vértice inicial (VI) dentro da janela e final (VF) fora dela.

Identificadas, cada uma destas situações, são seguidas por procedimentos específicos para identificar que parte do polígono está no interior da janela de recorte e, portanto, ser desenhada. A ideia é percorrer sequencialmente os lados do polígono a ser recortado e identificar os vértices necessários para desenhar a parte que está dentro da janela de recorte para então, visualizá-la. Isto pode ser feito construindo-se uma lista de vértices que descrevam o polígono recortado, ou seja, formada por vértices que estejam dentro da janela de visualização. Vamos aos procedimentos propostos por Sutherland e Hodgman para cada uma das quatro situações elencadas. Para análise, vamos considerar um vértice inicial (VI) e um vértice final (VF) para cada lado do polígono a ser recortado.

Situação (a):

Inicialmente veremos o procedimento para quando um lado do polígono está totalmente fora da janela de recorte, figura 85:

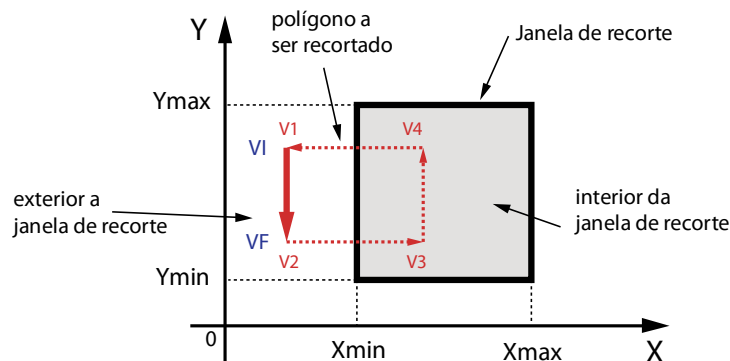


Figura 85: situação (a)

Tomemos o lado **V1V2** do polígono a ser recortado que para análise serão apelidados de **VIVF** (vértice inicial, vértice final). Podemos notar que o lado está totalmente fora da janela de recorte, pois temos:

VI : fora da janela de recorte;

VF : fora da janela de recorte.

Procedimento a ser adotado : nenhum, pois todo o lado do polígono se encontra fora da janela de recorte. Não se altera a lista que contém os novos vértices que representam o polígono recortado

Vamos analisar agora o lado V2V3 (VIVF). Podemos notar que ele está entrando na janela de recorte, figura 86:

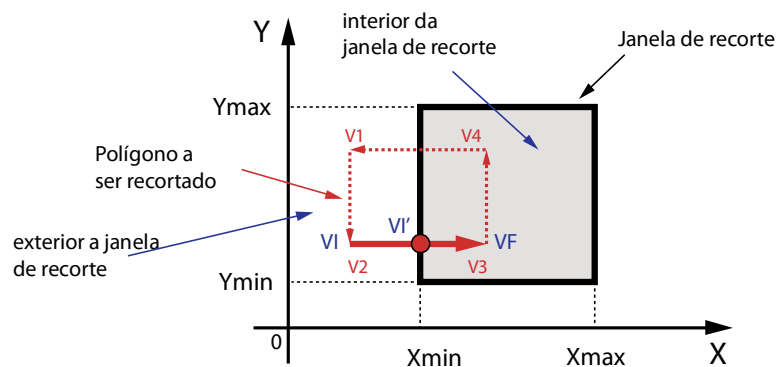


Figura 86: situação (b)

Temos:

VI : fora da janela de recorte;

VF : dentro da janela de recorte.

Procedimento a ser adotado: calcular a intersecção com a aresta da janela de recorte e inserir **VI'** e **VF** na lista de vértices que compõe o polígono recortado

Em seguida vamos analisar agora o lado **V3V4**. Podemos notar que ela está totalmente dentro da janela de recorte, figura 87:

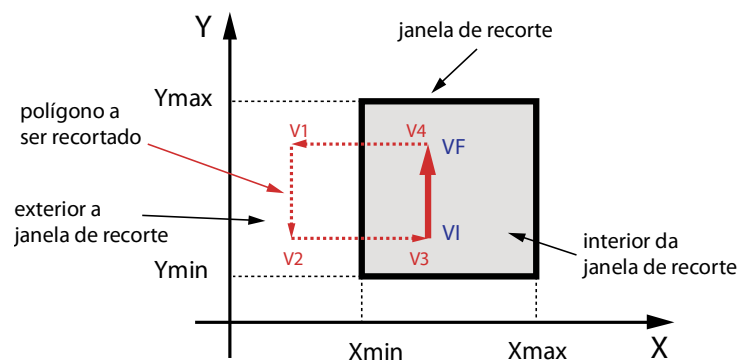


Figura 87: situação (c)

V3 : dentro da janela de recorte;

V4 : dentro da janela de recorte.

Procedimento a ser adotado: como os dois vértices que formam o lado do polígono estão dentro da janela de recorte vamos inserir na lista de vértices do polígono recortado apenas o vértice **VF**.

Finalmente temos a última situação elencada, ou seja, (d) lado *sainte*, possui vértice inicial dentro da janela e final fora dela, figura 88.

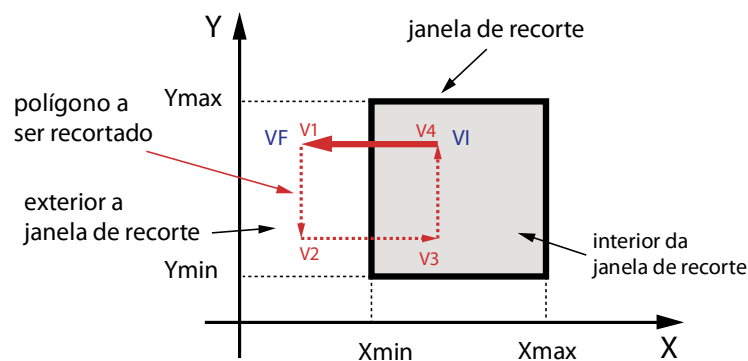


Figura 88: situação (d)

Analisando, temos:

VI: dentro da janela de recorte;

VF : fora da janela de recorte.

Procedimento a ser adotado: quando temos um lado do polígono como *sainte*, calculamos a intersecção **VF'** com a aresta da janela de recorte e inserimos na lista que representa o polígono recortado apenas a intersecção **VF'**.

Exemplo 12: vamos aplicar o método de **Sutherland e Hodgman** em um exemplo para ilustrar o seu funcionamento: Considere a figura 89.

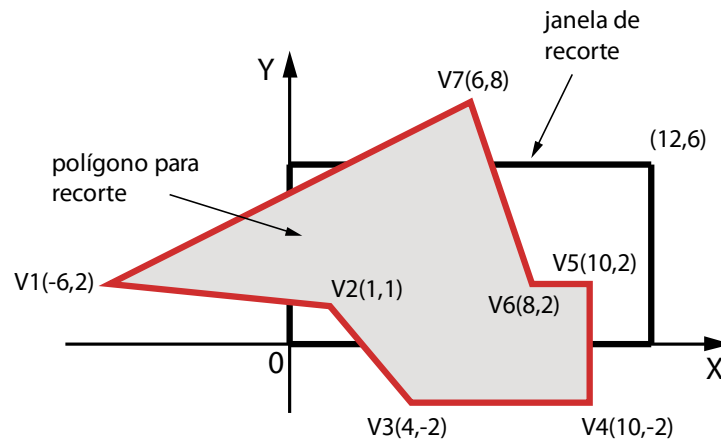


Figura 89: Polígono para recorte e Janela de recorte

Vamos começar pelo vértice **V1** e analisar o lado **V1V2** do polígono a ser recortado. Os outcodes de **V1** e **V2** são:

V1: **0001**

V2: **0000**

Como os outcodes não são simultaneamente **0000**, vamos realizar o **AND Lógico** para verificar a necessidade de recorte.

AND Lógico V1 V2: **0000**

O resultado do **AND lógico** indica a necessidade de fazer o recorte e de acordo com o outcode de **V2** o recorte deve ser feito calculando-se a intersecção com a aresta esquerda da janela de recorte. Além disso, os **outcodes** indicam que temos uma situação do tipo **(b) lado entrante**. Desta forma, vamos calcular a intersecção **V1'** e inserí-la na lista que desenha o o polígono recortado.

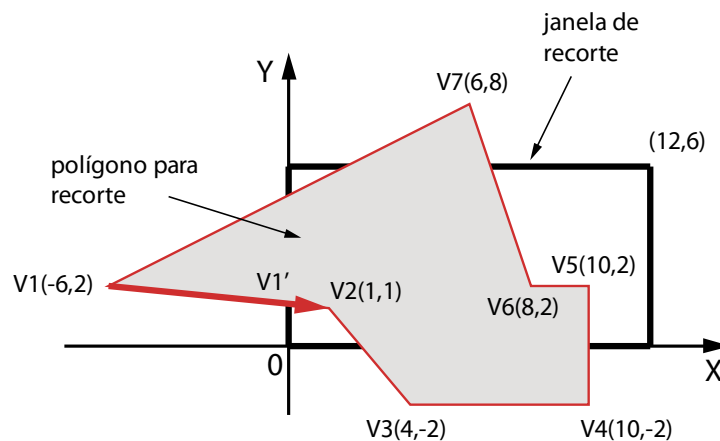


Figura 90: análise do lado **V1V2** em relação a janela de recorte

Cálculo de t da aresta esquerda (t_{ae}):

$$t_{ae} = \frac{x_{\min} - x_{V1}}{x_{V2} - x_{V1}}; \quad 0 \leq t_{ae} \leq 1$$

$$t_{ae} = \frac{0 - (-6)}{1 - (-6)} = \frac{6}{7}$$

$$x_{V1'} = 0$$

$$y_{V1'} = y_{V1} + (y_{V2} - y_{V1})t_{ae}$$

$$y_{V1'} = 2 + (1 - 2)\frac{6}{7} = \frac{8}{7}$$

$$V1' = (0, \frac{8}{7})$$

De acordo com o previsto para a situação **(b)** a lista que forma o polígono recortado recebe o ponto de intersecção **V1'** e o vértice final, ou seja, o vértice e **V2**.

Lista de Vértices - Polígono Recortado		
Vértice	X	Y
V1'	0	1.142
V2	1	1

Graficamente (figura 91):

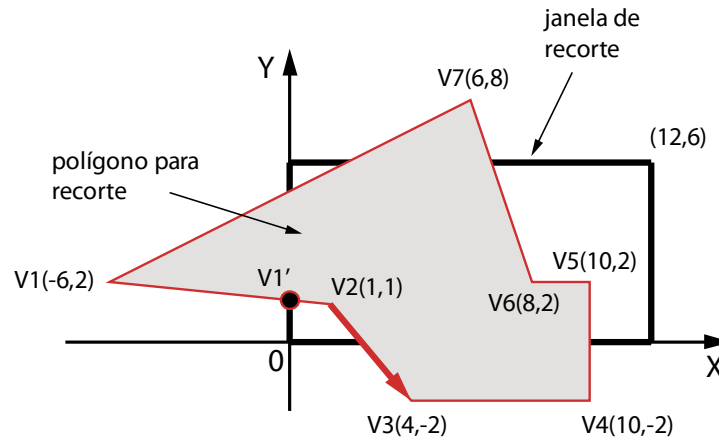


Figura 91: lado V2V3 - sainte

Vamos analisar o lado **V2V3** do polígono a ser recortado (figura 91). A verificação dos **outcodes** de **V2** e **V3** ficam:

VI: **0000**

VF: **0100**

Como os **outcodes** não são simultaneamente **0000**, vamos realizar o **AND Lógico** para verificar a necessidade de recorte.

AND Lógico V2V3: **0000**

O resultado do **AND lógico** indica a necessidade de fazer o recorte e de acordo com o **outcode** de **VF** o recorte deve ser feito calculando-se a intersecção com a **aresta inferior** da janela de recorte. Os **outcodes** indicam que temos uma situação do tipo **(d) lado sainte**. Desta forma, vamos calcular a intersecção e inserí-la na lista que desenha o polígono recortado.

Cálculo de t da aresta inferior (t_{ai}):

$$t_{ai} = \frac{y_{\min} - y_{V2}}{y_{V3} - y_{V2}}; \quad 0 \leq t_{ai} \leq 1$$

$$t_{ai} = \frac{0 - 1}{-2 - 1} = 1/3$$

$$x_{V3'} = x_{V2} + (x_{V3} - x_{V2})t_{ai}$$

$$y_{V3'} = y_{\min} = 0$$

$$x_{V3} = 1 + (4 - 1)(1/3) = 2$$

$$V3' = (2,0)$$

De acordo com o procedimento explicitado par a **situação do tipo (d)** devemos adicionar V3' na lista que compõe o polígono recortado que fica da seguinte forma:

Lista de Vértices - Polígono Recortado		
Vértice	X	Y
V1'	0	1.142
V2	1	1
V3'	2	0

Vamos agora analisar o lado **V3V4**. Mais uma vez vamos aos **outcodes** de **V3** e **V4** :

V3: **0100**

V4: **0100**

Como ambos não são simultaneamente **0000** temos que realizar o AND Lógico que fica:

And Lógico: **0100**

Como o **AND lógico** não é **0000**, significa que o lado esta totalmente fora da janela de recorte. Temos uma **situação do tipo (a)** cujo procedimento a ser adotado é não fazer anotações na lista que representa o polígono recortado. Passamos a analisar o próximo lado **V4V5**, cujos **outcodes** são:

V4: **0100**

V5: **0000**

Estes **outcodes** não são simultaneamente **0000** e por isso devemos analisar o AND Lógico que tem como resultado **0000**, o que indica necessidade de recorte. O **outcode** de **V4** indica que devemos calcular a intersecção com a aresta inferior da janela de recorte. Temos uma situação do tipo **(b) lado entrante**.

Cálculo de t da aresta inferior (t_{ai}):

$$t_{ai} = \frac{y_{min} - y_4}{y_5 - y_4}; \quad 0 \leq t_{ai} \leq 1$$

$$t_{ai} = \frac{0 - (-2)}{2 - (-2)} = 1/2$$

$$x_{V4'} = x_{V4} + (x_{V5} - x_{V4})t_{ai}$$

$$y_{V4'} = y_{min} = 0$$

$$x_{V4} = 10 + (10 - 10)(1/2) = 10$$

$$y_{V4'} = 0$$

$$V4' = (10,0)$$

Como temos uma **situação do tipo (b) entrante** devemos adicionar à lista que compõe o polígono recortado **V4'** e **V5**.

Lista de Vértices - Polígono Recortado		
Vértice	X	Y
V1'	0	1.142
V2	1	1
V3'	2	0
V4'	10	0
V5	10	2

Graficamente, (figura 92).

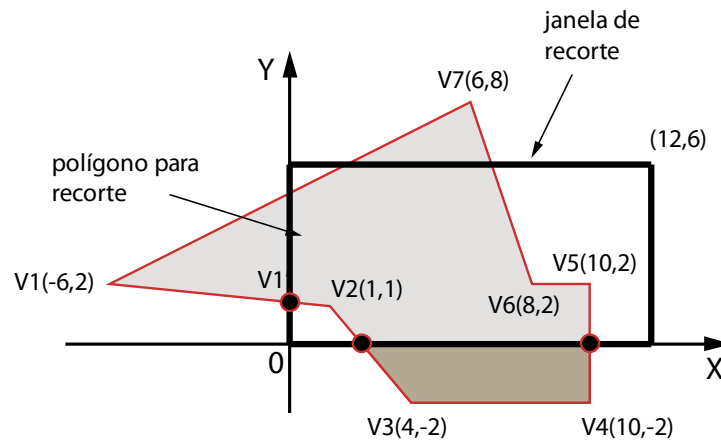


Figura 92: lado V2V3 - sainte

Vamos agora analisar o lado V5V6 cujos outcodes são:

V5: 0000

V6: 0000

Como os dois **outcodes** são simultaneamente 0000 indica que todo o segmento que representa o lado **V5V6** está dentro da janela de recorte. Portanto temos uma **situação do tipo (c)** que manda inserir na tabela de vértices do polígono recortado o vértice final, no caso **V6**.

Polígono Recortado		
Vértice	X	Y
V1'	0	1.142
V2	1	1
V3'	2	0
V4'	10	0
V5	10	2
V6	8	2

Vamos agora analisar o lado V6V7, cujos **outcodes** são:

V6: 0000

V7: 1000

Como os **outcodes** não são simultaneamente **0000** precisamos fazer o AND lógico, cujo resultado é **0000**, o que indica necessidade de recorte. O outcode do vértice **V7** indica uma intersecção com a aresta superior da janela de recorte. Temos uma **situação do tipo(d) sainte**. Vamos ao cálculo da intersecção V7' e inserí-la na lista de vértices do polígono recortado.

Cálculo de t para a aresta superior (t_{as}):

$$t_{as} = \frac{y_{máx} - y_{V6}}{y_{V7} - y_{V6}}; \quad 0 \leq t_{as} \leq 1$$

$$t_{as} = \frac{6 - 2}{8 - 2} = \frac{2}{3}$$

$$x_{V7'} = x_{V6} + (x_{V7} - x_{V6})t_{as}$$

$$y_{V7'} = y_{máx}$$

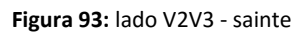
$$x_{V7'} = 8 + \frac{(6 - 8)2}{3} = \frac{20}{3}$$

$$y_{V7'} = 6$$

$$V7' = \left(\frac{20}{6}, 6\right)$$

Lista de Vértices - Polígono Recortado		
Vértice	X	Y
V1'	0	1.142
V2	1	1
V3'	2	0
V4'	10	0
V5	10	2
V6	8	2
V7'	6.66	6

Graficamente temos (figura, 93):



V1: 0001

O **outcode** de **V1** indica uma intersecção com a aresta esquerda da janela de recorte que pode ser calculada da seguinte forma:

$$t_{ae} = \frac{x_{min} - x_{V7}}{x_{V1} - x_{V7}}; \quad 0 \leq t_{ae} \leq 1$$

$$t_{ae} = \frac{0 - 6}{-6 - 6} = \frac{1}{2}$$

$$\begin{aligned} x_{V1'} &= x_{min} \\ y_{V1'} &= y_{V7} + (y_{V1} - y_{V7})t_{ae} \end{aligned}$$

$$\begin{aligned} x_{V1'} &= 0 \\ y_{V1'} &= 8 + \frac{(2 - 8)1}{2} = 5 \end{aligned}$$

$$V1' = (0,5)$$

Atualizando a tabela de vértices que integram o polígono recortado podemos ligar estes pontos para desenhá-lo.

Lista de Vértices - Polígono Recortado		
Vértice	X	Y
V1'	0	1.142
V2	1	1
V3'	2	0
V4'	10	0
V5	10	2
V6	8	2
V7'	6.66	6
V7''	2	6
V1''	0	5

O resultado, pode ser visto na figura 94. Gráficamente temos:

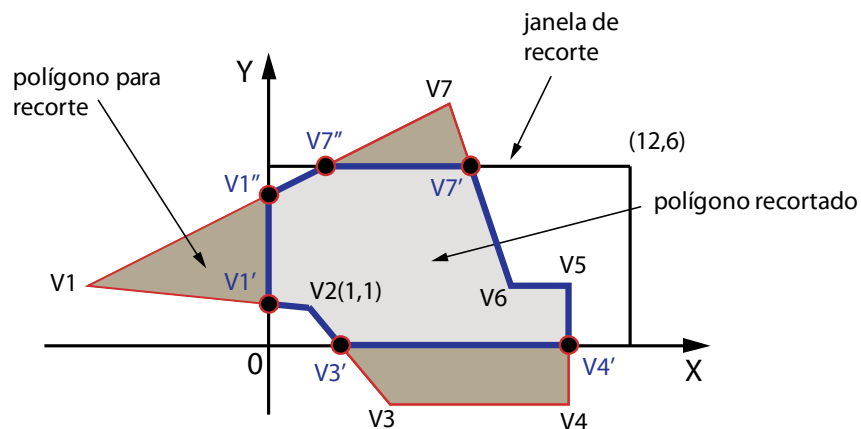


Figura 94: lado V2V3 - sainte

7.1.3.1.1 Síntese sobre o Método de Sutherland e Hodgman

O algoritmo proposto por de **Sutheland e Hodgman** para recorte de polígonos faz parte dos algoritmos clássicos de recorte. É simples de implementar e pode ser utilizado em janelas de recorte descritas por polígonos convexos, não apenas para retangulares. Pode ser generalizada para situações tridimensionais (Recorte 3D). Apresenta a desvantagem, de em alguns casos, com polígonos côncavos, ao ligar vértices gerados pela interseção com as arestas da janela de recorte, criar segmentos que sobrepõem arestas da janela e não fazem parte do polígono recortado, denominados de arestas fantasmas, que por sua vez, recebem este nome por serem irrelevantes ao ser desenho.

Bibliografia Recomendada

MAGALHÃES, L. P. **Computação Gráfica**: interfaces em sistemas de computação gráfica. Campinas: Unicamp, 1986.

PERSIANO, R. C. Marinho; OLIVEIRA, A. A. Fernandes de. **Introdução a computação gráfica**. Rio de Janeiro: LTC, 1988.

PLASTOCK, R. A.; HALLEY, G. **Computação Gráfica**. São Paulo: McGRAW-HILL, 1986.

ROGERS, D. F. **Procedural elements for computer graphics**. New York: McGRAW-HILL, 1985.

Cyrus, M.; Beck, J. Generalized two-and three-Dimensional Clipping. Computer and Graphics, 3, 23-28, 1978.

Cyrus, M., Beck, J.: Generalized Two and Three Dimensional Clipping, Computers&Graphics, Vol.3, No.1, pp.23-28, 1978