

# Visualização

Uéliton Freitas

Universidade Católica Don Bosco - UCDB

*freitas.ueliton@gmail.com*

9 de setembro de 2014

# Sumário

- 1 Introdução
- 2 Janela De Recorte
- 3 Algoritmos de Recorte de Primitivas 2D
  - Recorte de Pontos 2D
  - Recorte de Linhas 2D
  - Recorte de Polígonos 2D
  - Recorte de Outras Primitivas

# Introdução

## Visualização

- Determina quais partes serão mostradas na tela.

# Introdução

## Visualização

- Determina quais partes serão mostradas na tela.
- A imagem é determinada no **Sistema de Coordenadas do Mundo (World Coordinates)**, cujas partes selecionadas são mostradas no **Sistema de Coordenadas Locais (Local Coordinates)**.
  - Este processo envolve várias transformações: Rotação, Translações...
  - Operações para eliminar partes da imagem que estão fora da imagem de exibição.

# Introdução

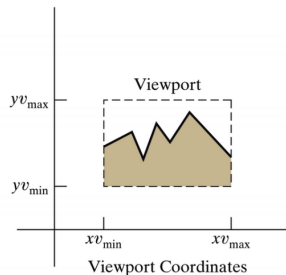
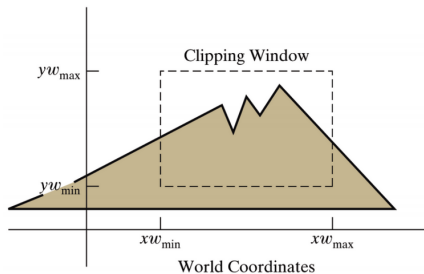
## Janela de Recorte ou Clipping Window

- Uma secção da cena é selecionada.
- O que estiver fora da cena não será mostrado.

## View Port

- A janela de recorte pode ser posicionada dentro de uma outra janela de recorte denominada **Viewport**.
  - Objetos dentro da **Janela de Recorte**(o que será visto) são posicionados dentro de uma **Viewport**, que por sua vez posiciona os objetos dentro da janela de sistema (onde serão visto).
  - **Múltiplas Viewports** podem ser utilizadas para mostrar várias partes de uma cena.

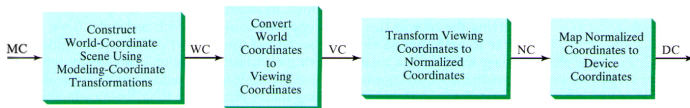
# Introdução



# Introdução

## Transformação da Visão

- Mapeamento do sistema de coordenadas do sistema de coordenadas mundo para o sistema de coordenadas de dispositivo.
- As coordenadas dos objetos são **normalizadas** entre 0 e 1 para acelerar o processo de recorte.



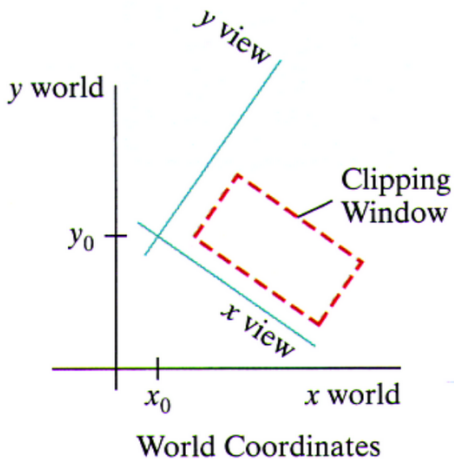
# Janela de Recorte

## Janela de Recorte

- Devido ao custo computacional as API's gráficas criam **Janelas de Recorte** em formato retangular com eixos  $x$  e  $y$ .
- Geralmente a **Janela de Recorte** é definida no sistema de coordenadas do mundo.
- Geralmente, a transformação de visão é definida em um sistema de coordenadas de visão dentro do sistema de coordenadas mundo.
  - Assim é possível determinar uma janela retangular em qualquer orientação.
  - Uma visão das coordenadas do mundo é obtida transformando a cena para as coordenadas de visão.



# Janela de Recorte



# Janela de Recorte

## Sistema De Coordenada da Janela de Recorte

- Escolhe-se um ponto  $\mathbf{P}_0 = (x_0, y_0)$  no sistema de coordenadas de visão e uma orientação obtida por meio de um vetor  $\mathbf{V}$  que orienta o eixo  $y_{view}$ .
  - $\mathbf{V}$  é chamado de **view-up vector**.

# Janela de Recorte

## Sistema De Coordenada da Janela de Recorte

- Escolhe-se um ponto  $\mathbf{P}_0 = (x_0, y_0)$  no sistema de coordenadas de visão e uma orientação obtida por meio de um vetor  $\mathbf{V}$  que orienta o eixo  $y_{view}$ .
  - $\mathbf{V}$  é chamado de **view-up vector**.
- Com o sistema de coordenadas de visão definido, são utilizadas translações e rotações para transformar as diferentes descrições de objetos para sobrepor os diferentes sistemas de coordenadas.
  - 1 Translada-se  $\mathbf{P}_0$  para a origem do sistema de coordenadas do mundo.
  - 2 O sistema é rotacionado para coincidir com o sistema de coordenadas do mundo.

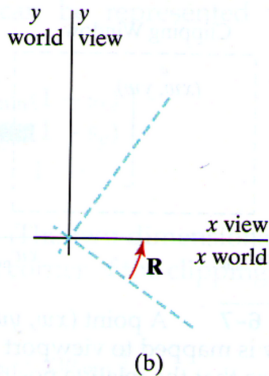
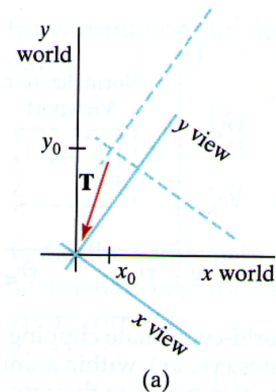
# Janela de Recorte

## Sistema De Coordenada da Janela de Recorte

- Escolhe-se um ponto  $\mathbf{P}_0 = (x_0, y_0)$  no sistema de coordenadas de visão e uma orientação obtida por meio de um vetor  $\mathbf{V}$  que orienta o eixo  $y_{view}$ .
  - $\mathbf{V}$  é chamado de **view-up vector**.
- Com o sistema de coordenadas de visão definido, são utilizadas translações e rotações para transformar as diferentes descrições de objetos para sobrepor os diferentes sistemas de coordenadas.
  - 1 Translada-se  $\mathbf{P}_0$  para a origem do sistema de coordenadas do mundo.
  - 2 O sistema é rotacionado para coincidir com o sistema de coordenadas do mundo.
- A conversão é dada por:

$$\mathbf{M}_{WC,VC} = \mathbf{R} \cdot \mathbf{T}$$

# Janela de Recorte



# Normalizações e Transformações

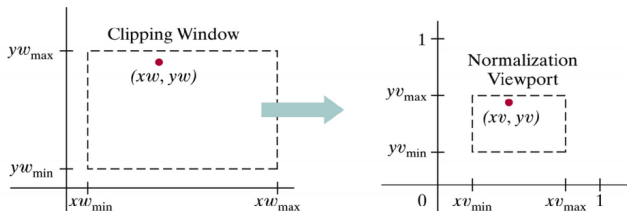
## Normalizações e Transformações da Viewport

- Em alguns sistemas as normalizações e transformações da **Window-Viewport** são feitas em uma mesma operação.
  - As coordenadas da viewport são definidas entre 0 e 1.
  - O quadrado unitário formado é então mapeado para o sistema de saída.
- Em outros sistemas, as normalizações e transformações são aplicadas antes das transformações da viewport.
  - Neste caso, as coordenadas da viewport são as coordenadas da tela.

# Normalizações e Transformações

## Mapeando uma Janela de Recorte para uma Viewport Normalizada

- Considerando uma viewport normalizada com valores entre 0 e 1, é necessário mapear a descrição de um objeto de modo que a sua posição seja mantida de acordo com a **janela de recorte**.



# Normalizações e Transformações

## Mapeando uma Janela de Recorte para uma Viewport Normalizada

- Para transformar um ponto no sistema de coordenadas do mundo para a viewport é necessário:

$$\frac{xv - xv_{min}}{xv_{max} - xv_{min}} = \frac{xw - xw_{min}}{xw_{max} - xw_{min}}$$
$$\frac{yv - yv_{min}}{yv_{max} - yv_{min}} = \frac{yw - yw_{min}}{yw_{max} - yw_{min}}$$

- Para as posições  $xv$  e  $yv$  temos:

$$xv = S_x \cdot xw + t_x$$

$$yv = S_y \cdot yw + t_y$$



# Normalizações e Transformações

## Mapeando uma Janela de Recorte para uma Viewport Normalizada

- Os fatores de escala são:

$$S_x = \frac{XV_{max} - XV_{min}}{XW_{max} - XW_{min}}$$

$$S_y = \frac{YV_{max} - YV_{min}}{YW_{max} - YW_{min}}$$

- Como fatores de translação temos:

$$t_x = \frac{XW_{max} \cdot XV_{min} - XW_{min} \cdot XV_{max}}{XW_{max} - XW_{min}}$$

$$t_y = \frac{YW_{max} \cdot YV_{min} - YW_{min} \cdot YV_{max}}{YW_{max} - YW_{min}}$$

# Normalizações e Transformações

## Mapeando uma Janela de Recorte para uma Viewport Normalizada

- Também é possível mapear o sistema de coordenadas do mundo para a viewport utilizando simples transformações.
  - Basta converter o retângulo da janela de recorte na viewport.
    - 1 Escala a janela de recorte para ter o tamanho da viewport usando o ponto fixo  $xw_{min}$  e  $yw_{min}$ .
    - 2 Translada  $xw_{min}$  e  $yw_{min}$  para  $xv_{min}$  e  $yv_{min}$

# Normalizações e Transformações

## Mapeando uma Janela de Recorte para uma Viewport Normalizada

- Termos como matriz de escala

$$\mathbf{S} = \begin{bmatrix} s_x & 0 & xW_{min}(1 - s_x) \\ 0 & s_y & yW_{min}(1 - s_y) \\ 0 & 0 & 1 \end{bmatrix}$$

- E a matriz de translação como sendo

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & xV_{min} - xW_{min} \\ 0 & 1 & yV_{min} - yW_{min} \\ 0 & 0 & 1 \end{bmatrix}$$

# Normalizações e Transformações

## Mapeando uma Janela de Recorte para uma Viewport Normalizada

- Termos como matriz composta

$$\mathbf{M}_{window,normview} = \mathbf{S} \cdot \mathbf{T}$$

- como sendo

$$\mathbf{M}_{window,normview} = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

- com  $s_x$ ,  $s_y$ ,  $t_x$  e  $t_y$  dados anteriormente.

# Normalizações e Transformações

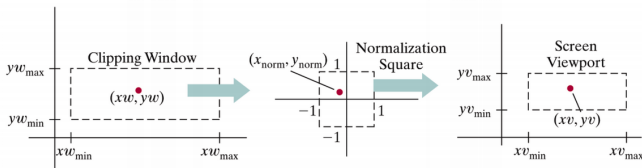
## Mapeando uma Janela de Recorte para uma Viewport Normalizada

- Neste mapeamento as posições relativas de cada objeto são mantidas.
  - Se um objeto aparece na janela de recorte o mesmo aparece na viewport.
- A proporção do objeto será mantida somente se a razão do aspecto da janela de recorte for a mesma da viewport.
  - Ou seja,  $s_x = s_y$ .

# Normalizações e Transformações

## Mapeando uma Janela de Recorte para um quadrado normalizado

- Uma outra abordagem, é transformar a janela de recorte em um quadrado normalizado, fazer os recortes da cena normalizada e então enviar para a viewport com sistema de coordenadas da tela.
- Nesta representação os objetos que não estão na cena (fora dos limites  $x = \pm 1$  e  $y = \pm 1$ ) são facilmente removidos da viewport.



# Normalizações e Transformações

## Mapeando uma Janela de Recorte para um quadrado normalizado

- Mapeando o conteúdo da janela de recorte para o quadrado normalizado, temos que estabelecer as seguintes coordenadas similarmente ao mapeamento window-viewport fazendo,

$$x_{vmin} = y_{vmin} = -1 \text{ e } x_{vmax} = y_{vmax} = 1$$

$$M_{window, normsquare} = \begin{bmatrix} \frac{2}{xW_{max} - xW_{min}} & 0 & -\frac{xW_{max} + xW_{min}}{xW_{max} - xW_{min}} \\ 0 & \frac{2}{yW_{max} - yW_{min}} & -\frac{yW_{max} + yW_{min}}{yW_{max} - yW_{min}} \\ 0 & 0 & 1 \end{bmatrix}$$

# Normalizações e Transformações

## Mapeando uma Janela de Recorte para um quadrado normalizado

- Após os algoritmos de corte serem aplicados o quadrado de tamanho 2 é então transformado em viewport,

$$x_{vmin} = y_{vmin} = 0 \text{ e } x_{vmax} = y_{vmax} = 1$$

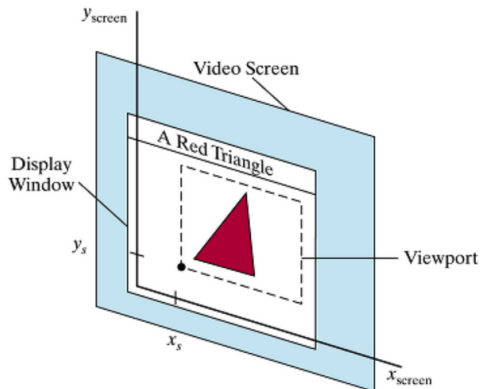
$$M_{normsquare, viewport} = \begin{bmatrix} \frac{xv_{max} - xv_{min}}{2} & 0 & \frac{xv_{max} + xv_{min}}{2} \\ 0 & \frac{yv_{max} - yv_{min}}{2} & \frac{yv_{max} + yv_{min}}{2} \\ 0 & 0 & 1 \end{bmatrix}$$



# Normalizações e Transformações

Mapeando uma Janela de Recorte para um quadrado normalizado

- Finalmente a viewport é posicionada na tela.



# Algoritmos de Recorte de Primitivas 2D

## Algoritmos de Recorte 2D

- Define o quais partes dos objetos aparecerão em uma cena.
- Identifica quais partes dos objetos estão fora da janela de recorte e elimina a descrição destes objeto no dispositivo de saída.
- Por eficiência, os recortes são aplicados nas **janelas de recorte normalizadas**.

# Algoritmos de Recorte de Primitivas 2D

## Algoritmos de Recorte 2D

- Existem vários algoritmos para o recorte de:
  - Pontos
  - Linhas
  - Áreas de Preenchimento(Polígonos).
  - Curvas
  - Texto
- Os três primeiros são padrões dos pacotes gráficos e apresentam maior rapidez de processamento caso os segmentos sejam retas.

# Algoritmos de Recorte de Primitivas 2D

## Algumas Definições

- Os recortes a seguir serão aplicados em uma janela de recorte retangular na posição padrão com arestas nas fronteiras  $xW_{min}$ ,  $xW_{max}$ ,  $yW_{min}$  e  $yW_{max}$ .
- Tipicamente correspondendo ao quadrado normalizado 0 e 1, ou -1 e 1.

# Algoritmos de Recorte de Primitivas 2D

## Recortes de Pontos 2D

- Dado um ponto 2D  $P = (x, y)$ , ele será apresentado no dispositivo de saída se, e somente se:

$$XW_{min} \leq X \leq XW_{max}$$

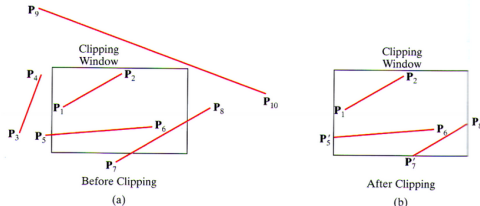
$$xy_{min} \leq y \leq yw_{max}$$

- Este processo é útil em recortes de partículas como nuvens, explosões, fumaça, etc.

# Algoritmos de Recorte de Primitivas 2D

## Recorte de Linhas 2D

- Processa cada linha de uma utilizando uma série de cálculos e intersecções para definir se a linha, ou parte dela será desenhada.
- A tarefa mais cara computacionalmente é o cálculo das intersecções.



# Algoritmos de Recorte de Primitivas 2D

## Recorte de Linhas 2D

- É fácil determinar se uma linha está completamente dentro da janela. Mas é um pouco mais complicado determinar se a mesma está fora.
  - Quando dois pontos da extremidade de uma linha ( $P_1$   $P_2$ ) estão dentro da janela, então a linha está dentro da janela e será desenhada.

# Algoritmos de Recorte de Primitivas 2D

## Recorte de Linhas 2D

- É fácil determinar se uma linha está completamente dentro da janela. Mas é um pouco mais complicado determinar se a mesma está fora.
  - Quando dois pontos da extremidade de uma linha ( $P_1$   $P_2$ ) estão dentro da janela, então a linha está dentro da janela e será desenhada.
  - Quando dois pontos da extremidade de uma linha estão fora de uma das fronteiras da janela (linha  $P_3$   $P_4$ ), a linha está completamente fora.



# Algoritmos de Recorte de Primitivas 2D

## Recorte de Linhas 2D

- É fácil determinar se uma linha está completamente dentro da janela. Mas é um pouco mais complicado determinar se a mesma está fora.
  - Quando dois pontos da extremidade de uma linha ( $P_1 P_2$ ) estão dentro da janela, então a linha está dentro da janela e será desenhada.
  - Quando dois pontos da extremidade de uma linha estão fora de uma das fronteiras da janela (linha  $P_3 P_4$ ), a linha está completamente fora.
  - Se ambos os testes falham, então a linha intersecta ao menos uma das fronteiras da janela, ou pode não cruzar o interior da mesma.

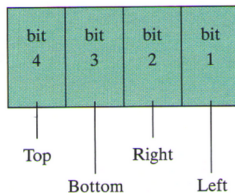
# Algoritmos de Recorte de Primitivas 2D

## Algoritmos de Recorte de Cohen-Sutherland

- Um dos primeiros algoritmos para acelerar o processo de recorte.
- O tempo de recorte é diminuído executando mais testes antes dos cálculos das intersecções.

## Algoritmo de Cohen-Sutherland

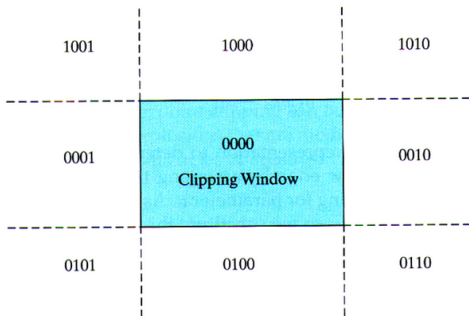
- Inicialmente, cada ponto final de uma linha é assinalado com um **código de região** constituído de 4 bits.



# Algoritmos de Recorte de Primitivas 2D

## Algoritmos de Recorte de Cohen-Sutherland

- As quatro fronteiras criam nove regiões de separação no espaço.
- Um ponto abaixo e a esquerda da janela recebe os valores 0101 e 000 caso esteja dentro da janela.



# Algoritmos de Recorte de Primitivas 2D

## Algoritmos de Recorte de Cohen-Sutherland

- Os valores dos bits são obtidos comparando as coordenadas  $(x, y)$  com as fronteiras de recortes.
  - O bit 1 é definido como 1 caso  $x < xw_{min}$ .
  - Os outros bits são obtidos de forma similar.

# Algoritmos de Recorte de Primitivas 2D

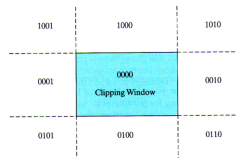
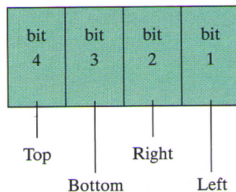
## Algoritmos de Recorte de Cohen-Sutherland

- Os valores dos bits são obtidos comparando as coordenadas  $(x, y)$  com as fronteiras de recortes.
  - O bit 1 é definido como 1 caso  $x < xW_{min}$ .
  - Os outros bits são obtidos de forma similar.
- É possível fazer esta comparação de forma mais eficiente seguindo dois passos:
  - 1 Calcula-se a diferença entre os pontos e as fronteiras da janela.
  - 2 Usar o sinal resultante para definir o valor do código (- corresponde a 1, + corresponde a 0).
    - bit 1 é o sinal  $x - xW_{min}$ .
    - bit 2 é o sinal  $xW_{max} - x$ .
    - bit 3 é o sinal  $y - yW_{min}$ .
    - bit 4 é o sinal  $yW_{max} - y$ .

# Algoritmos de Recorte de Primitivas 2D

# Algoritmos de Recorte de Cohen-Sutherland

- Assim é possível determinar facilmente se uma linha está totalmente dentro ou fora da janela.
  - Linhas totalmente dentro possuem seu pontos com valores 0000.
  - Linhas com valores 1 na mesma posição estão totalmente fora da janela.
    - Uma linha com pontos finais indicando os valores 1001 e 0101 está completamente a esquerda da janela de recorte.



# Algoritmos de Recorte de Primitivas 2D

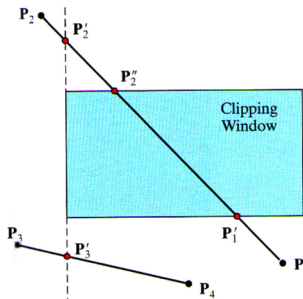
## Algoritmos de Recorte de Cohen-Sutherland

- Os testes podem ser executados de forma bastante eficiente utilizando operações lógicas simples:
  - Quando a operação **ou** entre os bits for false (0000), a linha está completamente dentro.
  - Quando a operação **e** for verdadeira (não 0000) a linha está completamente fora.

# Algoritmos de Recorte de Primitivas 2D

## Algoritmos de Recorte de Cohen-Sutherland

- As linhas que não há a certeza se estão completamente dentro ou fora da janela são então processadas para ver se há intersecção com a janela.





# Algoritmos de Recorte de Primitivas 2D

## Algoritmos de Recorte de Cohen-Sutherland

- De acordo com a intersecção de cada linha com as fronteiras da janela, a linha é recortada até que sobre apenas o que está dentro da janela, ou nenhuma parte esteja dentro da mesma.

# Algoritmos de Recorte de Primitivas 2D

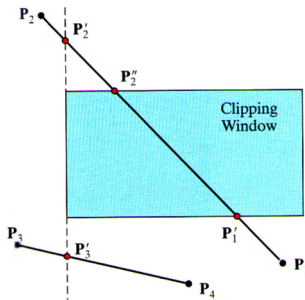
## Algoritmos de Recorte de Cohen-Sutherland

- De acordo com a intersecção de cada linha com as fronteiras da janela, a linha é recortada até que sobre apenas o que está dentro da janela, ou nenhuma parte esteja dentro da mesma.
- Para Determinar se a linha cruza a fronteira, basta verificar se o bit da fronteira dos pontos finais variam.
  - Se um dos bits for 0 e o outro 1, a linha cruza a fronteira.

# Algoritmos de Recorte de Primitivas 2D

## Verificando a fronteira esquerda

- $P_1 = 0100$  - Está dentro da fronteira esquerda.
- $P_2 = 1001$  - Está fora da fronteira esquerda.
  - Calcula a intersecção  $P'_2$  e recorta a linha  $\overline{P_1 P'_2}$



# Algoritmos de Recorte de Primitivas 2D

## Intersecção de uma Linha com a Janela

- Para determinar a intersecção de uma linha com pontos  $(x, y)$  e  $(x', y')$  podemos usar a equação:

$$y' = y + m(x' - x) \quad (1)$$

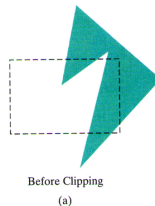
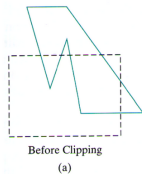
$$x' = x + \frac{y' - y}{m} \quad (2)$$

- O valor de  $x'$  será  $xw_{min}$  ou  $xw_{max}$  na Equação 1, e o valor de  $y'$  será  $yw_{min}$  ou  $yw_{max}$  na Equação 2. A inclinação será  $m = \frac{y' - y}{x' - x}$

# Algoritmos de Recorte de Primitivas 2D

## Recorte de Polígonos 2D

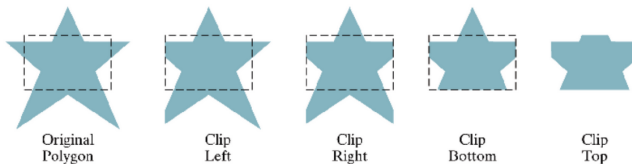
- Para fazer os recortes de polígonos, o algoritmo de recorte de linhas não pode ser utilizado porque, em geral, não gerariam polígonos fechados.
  - Produziriam linhas desconexas sem informações de como uni-las.



# Algoritmos de Recorte de Primitivas 2D

## Recorte de Polígonos 2D

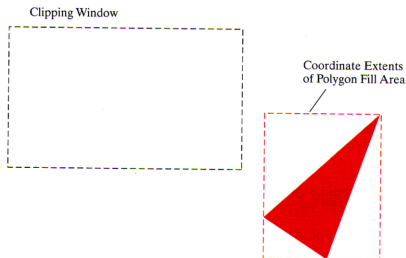
- É possível fazer o recorte de polígonos de forma semelhante ao recorte de linhas.
  - Isto é feito formando um novo polígono cada vez que o recorte de uma fronteira é feito.



# Algoritmos de Recorte de Primitivas 2D

## Recorte de Polígonos 2D

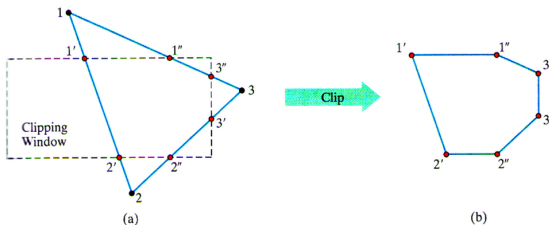
- Para verificar se um polígono está completamente dentro da janela, basta verificar se suas coordenadas máximas e mínimas estão dentro da janela de recorte.
- Quando uma área não pode ser verificada como totalmente dentro da janela, as intersecções são calculadas.



# Algoritmos de Recorte de Primitivas 2D

## Recorte de Polígonos 2D

- Uma forma simples de se fazer os recortes é criando uma nova lista de vértices cada vez que são feitos recortes em uma fronteira da janela, e então passar a nova lista de vértices para a nova fronteira.
- **Polígonos Côncavos** podem gerar várias listas de vértices.





# Algoritmos de Recorte de Primitivas 2D

## Algoritmo de Sutherland-Hodgman

- Uma forma eficiente de efetuar os recortes é enviar cada vértice do polígono para cada estágio de recorte de forma que os vértices recortados possam ser enviados diretamente para o próximo estágio.
  - Elimina a necessidade de uma lista de vértices.
  - Há possibilidade de efetuar os recortes em paralelo.

# Algoritmos de Recorte de Primitivas 2D

## Algoritmo de Sutherland-Hodgman

- A principal estratégia deste algoritmo é mandar os pares de pontos finais de cada linha sucessiva do polígono para uma série de recortadores (esquerdo, direito, inferior e superior).
- Conforme o recorte é executado para um par de vértice, as coordenadas recortadas são enviadas para o próximo recortador.

# Algoritmos de Recorte de Primitivas 2D

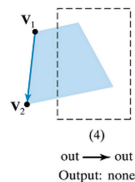
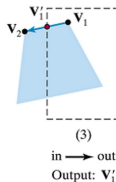
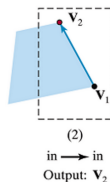
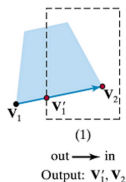
## Algoritmo de Sutherland-Hodgman

- Há quatro casos que precisam de atenção nos recortes de arestas de um polígono.
  - 1 O primeiro ponto final está dentro da janela de recorte e o segundo fora.
  - 2 Ambos os pontos finais estão dentro da janela de recorte.
  - 3 O primeiro ponto final da aresta está dentro da janela de recorte e o segundo está fora.
  - 4 Ambos os pontos finais estão fora da janela de recorte.

# Algoritmos de Recorte de Primitivas 2D

## Algoritmo de Sutherland-Hodgman

- Para facilitar a passagem de um vértice de um recorte para outro, a saída de cada recortador pode ser da seguinte forma:



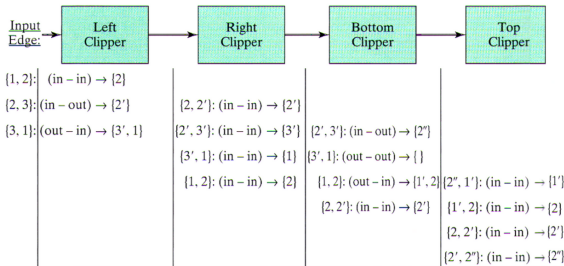
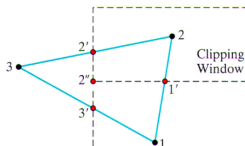
# Algoritmos de Recorte de Primitivas 2D

## Algoritmo de Sutherland-Hodgman

- Conforme cada par de vértice sucessivos é passado para cada um dos recortadores, a saída é gerada de acordo com os seguintes passos:
  - 1 Se o primeiro vértice está fora da janela e o segundo está dentro, a intersecção obtida é enviada juntamente com o segundo vértice para o próximo recorte.
  - 2 Se ambos os vértices estão dentro, somente o segundo vértice é enviado.
  - 3 Se o primeiro vértice está dentro da janela e o segundo está fora, é enviado para o próximo recortador somente a intersecção.
  - 4 Se ambos os vértices estão fora, nada é enviado.

## Recorte de Polígonos 2D

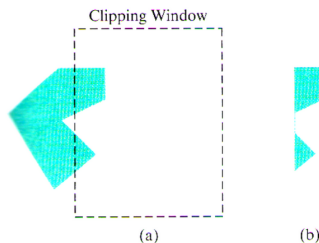
## Algoritmos de Recorte de Primitivas 2D



# Algoritmos de Recorte de Primitivas 2D

## Limitações

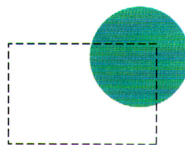
- Para polígonos côncavos este algoritmo causa problemas já que ele gera apenas uma lista de vértices.
- Uma solução seria dividir o polígono côncavo em partes convexas.



# Algoritmos de Recorte de Primitivas 2D

## Recortes de Curvas

- As curvas podem ser recortadas com as abordagens apresentadas anteriormente.
  - Se as curvas forem aproximações poligonais, é utilizado o algoritmo descrito anteriormente.
  - Caso contrário o procedimento envolve equações não lineares.



Before Clipping



After Clipping