

Recorte

- Finalidade: eliminar objetos que não aparecem na Windows
- Redefinir objetos que aparecem parcialmente na Windows
- Algoritmos:
 - recorte de pontos
 - recorte de linhas
 - Algoritmo de *Cohen-Sutherland*
 - Algoritmo de subdivisão - ponto médio
 - Algoritmo de *Sutherland-Hodgman*
 - Algoritmo de *Weiler-Atherton*

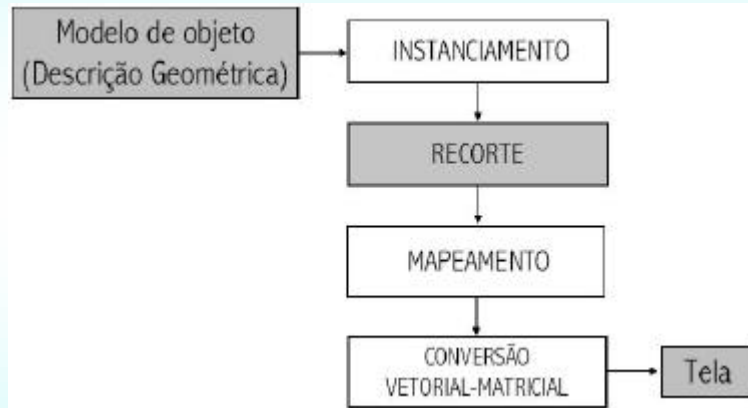
Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Recorte



Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Recorte



Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Recorte de pontos

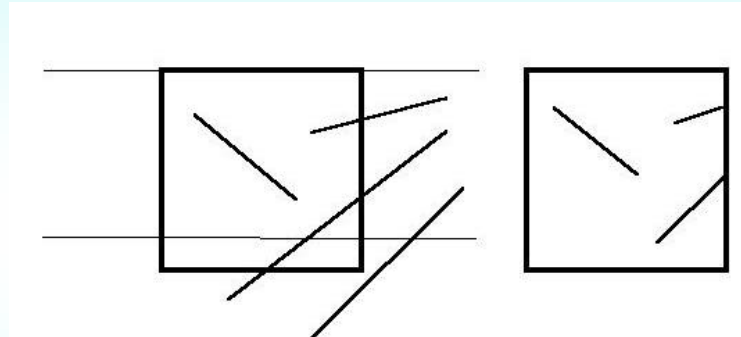
- Processo rápido e simples
- O ponto só será apresentado na viewport se:

$$x_{\min} \leq x \leq x_{\max} \quad \text{e} \quad y_{\min} \leq y \leq y_{\max}$$

- O ponto que não satisfaz todas as inequações não pode ser apresentado na viewport

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Recorte de Linhas



- processo exige mais cálculos e testes
- considerar pontos finais das linhas: estratégia

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Recorte de Linhas

- 1ª Solução:
 - checagem, utilizando equação paramétrica da reta
 - exige demasiada quantidade de cálculos e testes
 - estratégia: fazer testes iniciais de forma a detectar se cálculos de interseções são necessários
 - linhas aceitas ou rejeitadas trivialmente, checando pontos finais

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Recorte - Linhas

if P1 = DENTRO and P2 = DENTRO

~ Desenha linha P1->P2

if P1 = DENTRO and P2 = FORA or

P1 = FORA and P2 = DENTRO

~ Acha interseção da linha com a window (qual a borda?)

~ Redefine P2 (ou P1)

~ Desenha linha P1->P2 (ou P2->P1)

if P1 = FORA and P2 = FORA

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

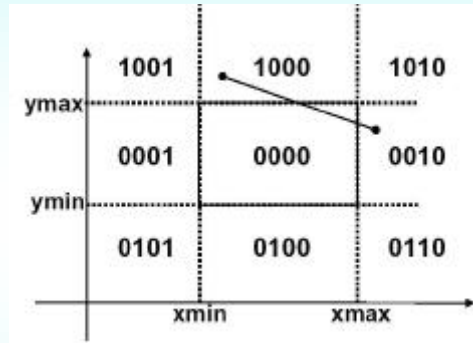
Algoritmo de Cohen Sutherland

- Identifica, de forma eficiente, que linhas são trivialmente aceitas ou rejeitadas, por meio de regiões
- dispensa cálculos de interseções nos casos de linhas:
 - aceitas trivialmente
 - rejeitadas trivialmente

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Algoritmo de Cohen Sutherland

Mapa de bits:
(b3 b2 b1 b0)



- 4º bit - o ponto está à esquerda da janela - b0 - P(4)
- 3º bit - o ponto está à direita da janela - b1 - P(3)
- 2º bit - o ponto está à abaixo da janela - b2 - P(2)
- 1º bit - o ponto está à acima da janela - b3 - P(1)

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Algoritmo de Cohen Sutherland

- 1º Passo: associar códigos aos pontos finais, usando a regra:

$$\begin{cases} \text{se } x_1 < X_L \Rightarrow P_{\text{code}(4)} = 1; \text{ do contrário: } P_{\text{code}(4)} = 0 \\ \text{se } x_1 > X_L \Rightarrow P_{\text{code}(3)} = 1; \text{ do contrário: } P_{\text{code}(3)} = 0 \\ \text{se } y_1 < Y_T \Rightarrow P_{\text{code}(1)} = 1; \text{ do contrário: } P_{\text{code}(1)} = 0 \\ \text{se } y_1 > Y_B \Rightarrow P_{\text{code}(2)} = 1; \text{ do contrário: } P_{\text{code}(2)} = 0 \end{cases}$$

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Algoritmo de Cohen Sutherland

■ 2º Passo: verificar se a linha é totalmente visível:

- se os dois códigos associados às duas extremidades do segmento de reta forem zero: linha totalmente visível

```
soma1 := 0 ; soma2 := 0 ;  
for i = 1 to 4  
    soma1 := soma1 + P1CODE (i)  
    soma2 := soma2 + P2CODE (i)  
if soma1 = 0 and soma2 = 0 then  
    linha totalmente visível
```

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Algoritmo de Cohen Sutherland

■ 3º Passo: verificar se a linha é invisível, estando totalmente à direita, esquerda, acima ou abaixo da janela:

```
Inter := 0  
for i = 1 to 4  
    Inter := Inter + integer ((P1.  
    CODE(i) + P2. CODE(i)) / 2)  
if Inter <> 0 then linha invisível  
    else next i
```

Ex: { linha KL P1 (0 0 1 0) ; P2(0 1 1 0) ⇒ OK - linha invisível
linha GH P1(0 0 0 1) ; P2(1 0 0 0) ⇒ Falha
linha IJ P1 (1 0 0 1) ; P2 (1 0 0 0) ⇒ OK - linha invisível

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Algoritmo de Cohen Sutherland

- 4º Passo: verificar se um dos pontos está dentro da janela:

- soma1 = 0 ou soma2 = 0
- Se estiver, basta tomar uma interseção:

$$X_c \leq X_{c'} \leq X_d$$

$$X_e \leq X_{f'} \leq X_f$$

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

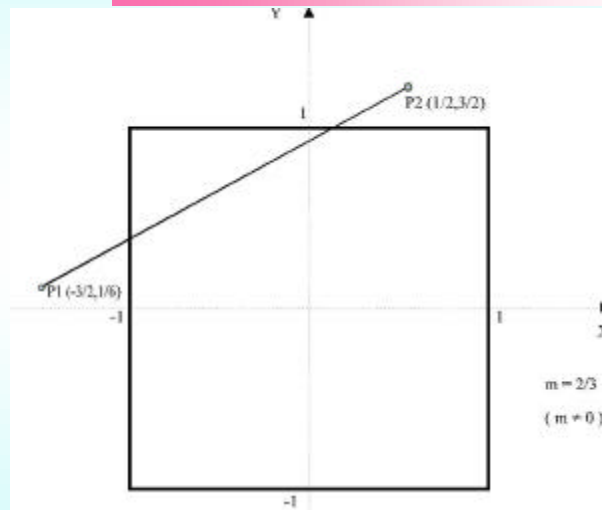
Algoritmo de Cohen Sutherland

- 5º Passo: A linha é totalmente visível ou totalmente invisível. Se visível, calcular as interseções da mesma com as bordas:

Esquerda: $X_L, Y = M * (X_L - X_1) + Y_1 ; M0;$
Direita: $X_R, Y = M * (X_R - X_1) + Y_1 ; M0;$
Top: $Y_T, X = X_1 + 1/M * (Y_T - Y_1); M0;$
Botton: $Y_B, X = X_1 + 1/M * (Y_B - Y_1); M0;$

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Exemplo



Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Algoritmo de subdivisão por ponto médio

■ Motivação:

- Algoritmo anterior necessitava de cálculos para obtenção da interseção da linha com limites da janela
- usar a busca binária de forma a evitar o cálculo citado

■ Caso particular do Algoritmo anterior, proposto por Sproull e Sutherland

■ implementação em hardware apresenta melhor desempenho:

- uso de arquitetura paralela para divisão e multiplic.

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Algoritmo de subdivisão por ponto médio

■ Divisão por 2, em bits:

- nº 6: 0110, deslocando 1 bit para a direita:
0011: nº 3

■ Estratégia:

- um teste inicial é aplicado para detectar linhas trivialmente aceitas ou rejeitadas
- linhas para as quais o teste inicial falha, são subdivididas em 2 partes iguais:
 - $X_m = (x_1 + x_2) / 2$
 - $Y_m = (y_1 + y_2) / 2$

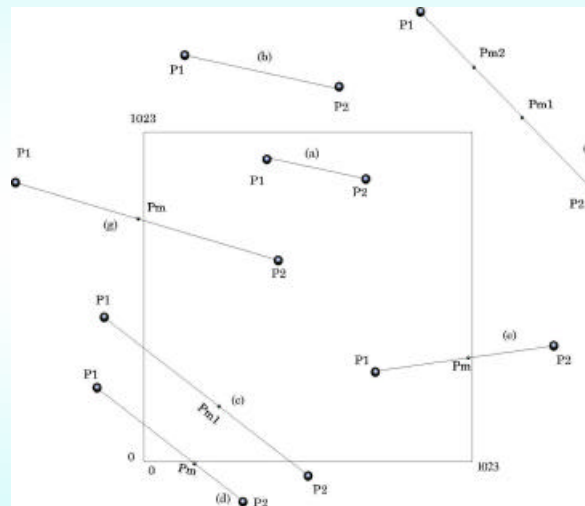
Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Algoritmo de subdivisão por ponto médio

- O teste é, então, aplicado a cada uma das metades até a obtenção da interseção com as bordas da janela - ou seja, até que o comprimento da parte resultante seja *infinitesimal*.
- A visibilidade do ponto é, então, determinada - busca logarítmica

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Algoritmo de subdivisão por ponto médio



Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Algoritmo de subdivisão por ponto médio

Linha 'f': P1-1000; P2-0010 necessário checar

Subdivisão P_{m1} -1010 P2-0010 trivialmente rejeitado

Subdivisão P_{m1} - 1010 P1-1000 trivialmente rejeitado

Subdivisões sucessivas: rejeitar a linha

Linha 'c': P1 (0001) P2 (0100) necessário checar ponto médio P_{m1} (0001) mesmo resultado para ambos os lados

* Analisando P_{m1} P2 inicialmente:

$$\text{dividimos em } P_{m2} \Rightarrow \begin{cases} P_{m1}(0000) \\ P_{m2}(0000) \end{cases} \Rightarrow \text{totalmente visível}$$

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Algoritmo de subdivisão por ponto médio

$$\begin{cases} P_{m2}(0000) \\ P_2(0100) \end{cases} \Rightarrow \text{parcialmente visível}$$

- Traça-se $P_{m1}P_{m2}$; continuamos a divisão de $P_{m2}P_2$
- divisões sucessivas: segmentos menores serão obtidos e ao final, teremos a obtenção da interseção com a janela
- O segmento $P_{m1}P_1$ sofrerá a mesma análise
- Ideal: obter os dois pontos e traçar a linha entre eles.

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

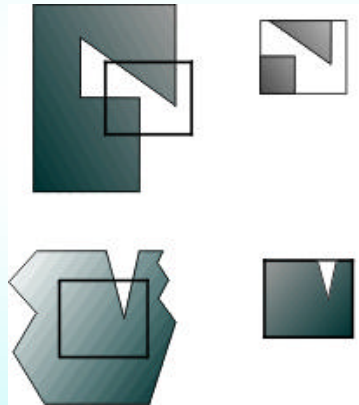
Recorte de Polígonos

- Vértices do polígono: armazenados em uma estrutura de dados conveniente
- exibição do polígono:
 - operação de transformação de visualização
 - recorte
 - conversão nas coordenadas do equipamento

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Recorte de Polígonos

- Algoritmo deve ser capaz de identificar situações distintas:



Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

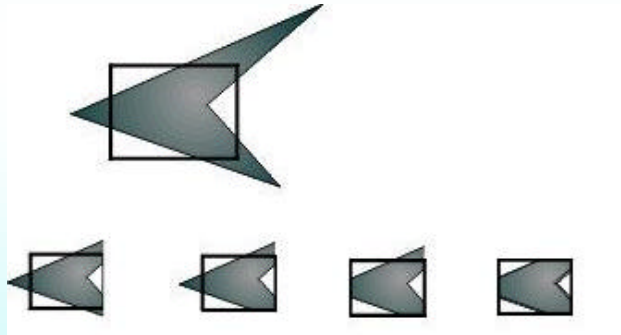
Algoritmo de Sutherland-Hodgman

- Marco no desenvolvimento da Computação Gráfica
- Até então: recorte de polígonos usava estratégias do recorte de retas:
 - cada aresta era analisada em relação à região de recorte como um todo
 - perda da conectividade do polígono recortado
- Só funciona para regiões convexas

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Algoritmo de Sutherland-Hodgman

- Estrat gia: recortar o pol gono atrav s do recorte de suas laterais. Ser o efetuados quatro recortes:



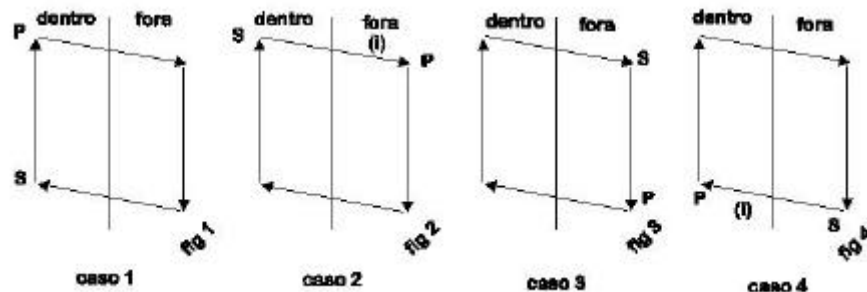
Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Sutherland - Hodgman

- Algoritmo:
 - suponha um pol gono de lados dados por v rtices: v_1, v_2, \dots, v_n ;
 - para cada lado, observa-se a rela  o entre v rtices sucessivos e as janelas (limites)
 - lados definidos pelos v rtices da lista de sa das ser o apresentados na tela.

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Sutherland - Hodgman



CASO 1: um dos dois vértices é adicionado à lista de saídas (p, no caso)

CASO 2: o ponto “i” de interseção é tratado como um vértice de saída (a ser traçado)

CASO 3: os dois vértices são descartados.

CASO 4: os dois pontos “i” e “p” são colocados na lista de vértices de saída.

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Sutherland - Hodgman

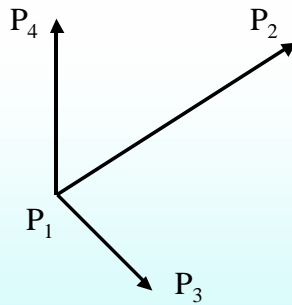
■ Algoritmo de obtenção das interseções:

- O primeiro ponto não é colocado na lista de saídas (s, da fig. 1, anterior), já que o mesmo é o vértice inicial e já se encontra na mesma, pois o processo é seqüencial;
- para linhas do polígono totalmente invisível, nenhum ponto é adicionado à lista de saídas (caso 3);
- para casos 2 e 4, é necessário calcular as interseções;

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Sutherland - Hodgman

- para um dado vértice, se necessário calcular se o mesmo está dentro ou fora de uma janela, temos uma função que aplica um teste baseado em produto vetorial, como:



$$P_1P_2 \otimes P_1P_3 = \otimes$$

$$P_1P_2 \otimes P_1P_4 = \odot$$

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Sutherland - Hodgman

■ Resultados:

- vetor entrando: módulo negativo - o ponto está do lado de dentro da janela
- vetor saindo: módulo positivo - o ponto está do lado de fora da janela

■ Módulo

- sejam $v(v.x, v.y)$ e $w(w.x, w.y)$ os vetores, o produto vetorial é um vetor de magnitude:

$$\blacksquare v.x \cdot w.y - v.y \cdot w.x$$

- ✓ positivo: ponto fora, negativo: ponto dentro.

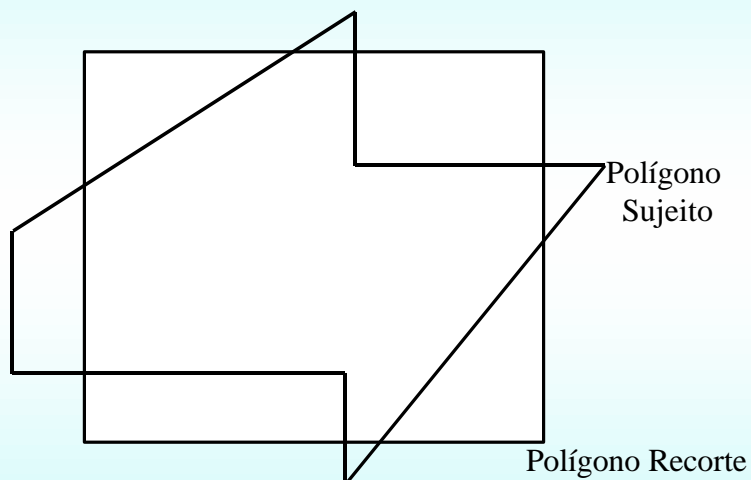
Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Weiler Atherton

- Solução para qualquer tipo de região, não só as convexas
- mais complexo que o anterior
- mais poderoso
- Dois polígonos:
 - polígono a ser recortado: polígono sujeito
 - polígono que é definido na região de recorte: polígono a ser recortado.
 - região de recorte: polígono recorte

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Weiler Atherton



Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

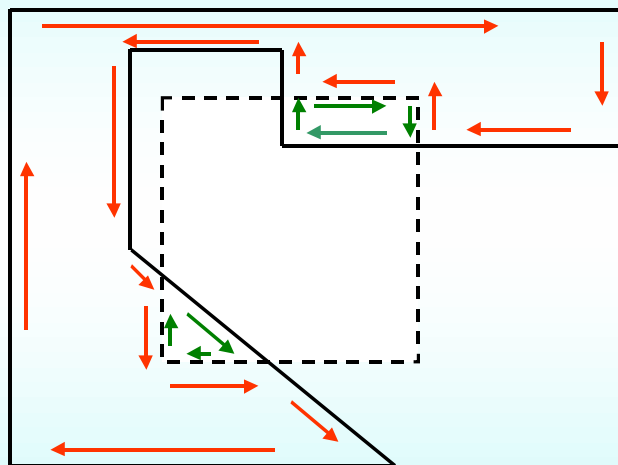
Weiler - Atherton

■ Estratégia:

- descreve os dois polígonos por uma lista de vértices
- percorre o polígono sujeito, na direção horária, até obter uma interseção com o polígono recorte
- se o movimento indica entrar no polígono recorte, o algoritmo passa a varrer limites do polígono sujeito
- se o movimento indica sair do polígono recorte, uma varredura, por retorno à direita é feita, seguindo o polígono recorte no sentido horário até obter a interseção que originou o movimento

Prof. Dr. Alexandre Cardoso - alexandre@ufu.br

Weiler - Atherton



Prof. Dr. Alexandre Cardoso - alexandre@ufu.br