

Clipping (Recorte)

Traina A., Oliveira, M. C. F; Apostial de Computação Gráfica , 2006. Disponível em : <http://www.inf.ufes.br/~thomas/graphics/> Data de acesso: 12/3/2015. Originalmente extraída de: <http://www.gbdi.icmc.usp.br/documentacao/apostilas/cg/downloads/modelagem.pdf>

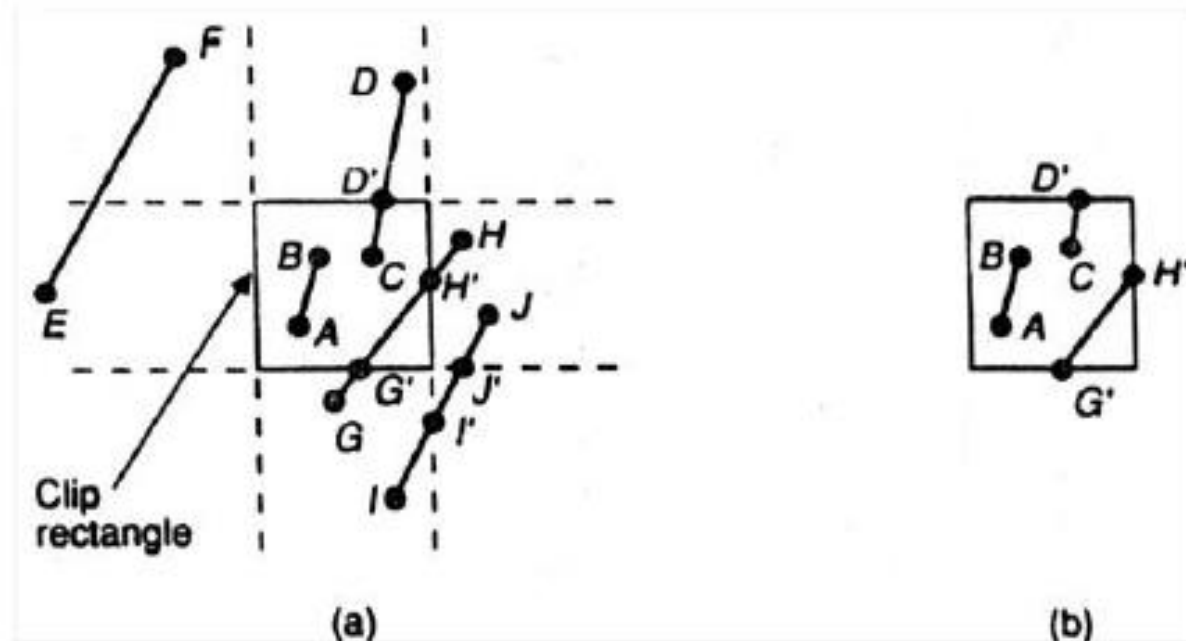
Clipping

- Um “pacote gráfico” atua como intermediário entre o aplicativo (e o seu modelo/estrutura de dados interna) e o hardware de visualização. Sua função é aproximar primitivas matemáticas (“ideais”), descritas em termos de vértices num sistema de coordenadas cartesianas, por conjuntos de pixels com a cor ou nível de cinza apropriado. Estes pixels são armazenados num bitmap na memória da CPU, ou num frame buffer (memória de imagem no controlador do dispositivo).

“Clipping”

- O “recorte” é necessário para que a imagem “apareça” dentro do retângulo de visualização definido para ela.
- Uma técnica óbvia é recortar a primitiva antes da conversão matricial, calculando analiticamente suas intersecções com o retângulo de recorte/ visualização[clip rectangle]. Estes pontos de intersecção são então usados para definir os novos vértices para a versão recortada da primitiva.

Recorte de Segmento de Reta



Recorte de Pontos Extremos

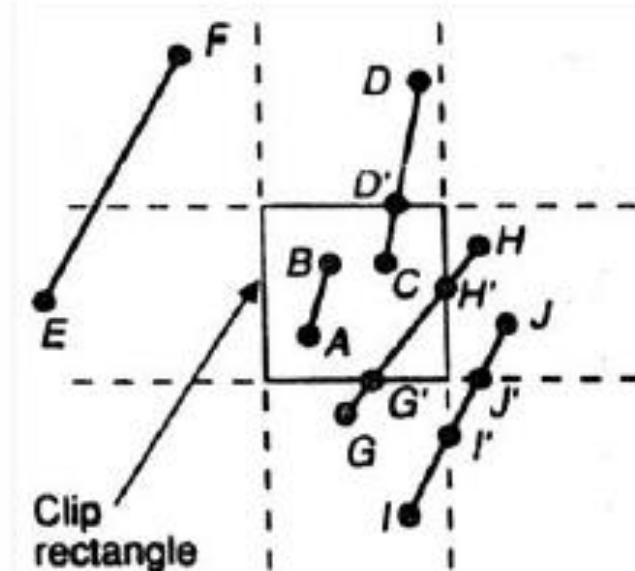
- Se a fronteira do retângulo de recorte tem coordenada x no intervalo x_{\min} e x_{\max} , e coordenada y no intervalo y_{\min} e y_{\max} , então 4 desigualdades precisam ser satisfeitas para que um ponto (x, y) esteja dentro do retângulo de recorte:

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

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

Algoritmo de Cohen-Sutherland

- Primeiramente, verifica pares de pontos extremos.
- Se a linha não pode ser trivialmente aceita, são feitas verificações por regiões.
- Segmento EF
- Da mesma forma, podemos rejeitar trivialmente linhas com ambos os extremos em regiões à direita de x_{max} , abaixo de y_{min} e acima de y_{max} .

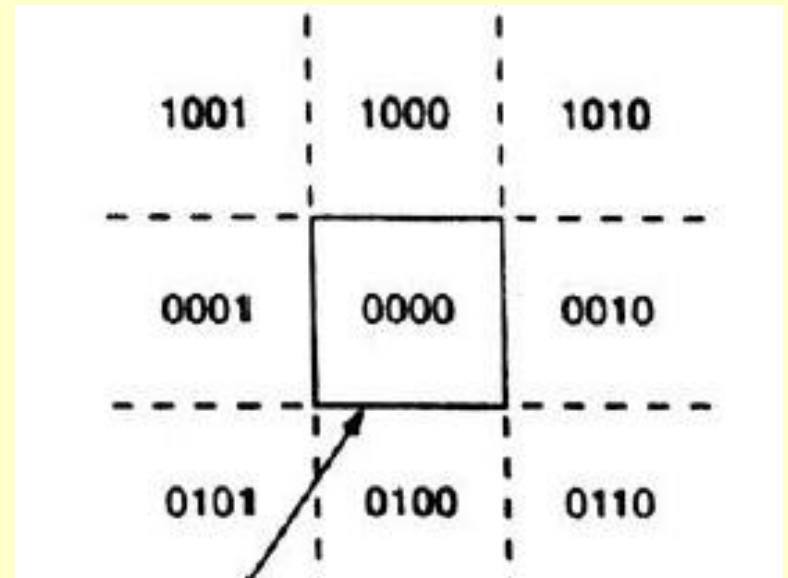


Algoritmo de Cohen-Shutterland

- Se o segmento não pode ser trivialmente aceito ou rejeitado, ele é subdividido em dois segmentos por uma aresta de recorte, um dos quais pode ser trivialmente rejeitado. Dessa forma, um segmento é recortado *iterativamente* testando-se aceitação ou rejeição trivial, sendo subdividido se nenhum dos dois testes for bem-sucedido, até que o segmento remanescente esteja totalmente contido no retângulo ou totalmente fora dele.

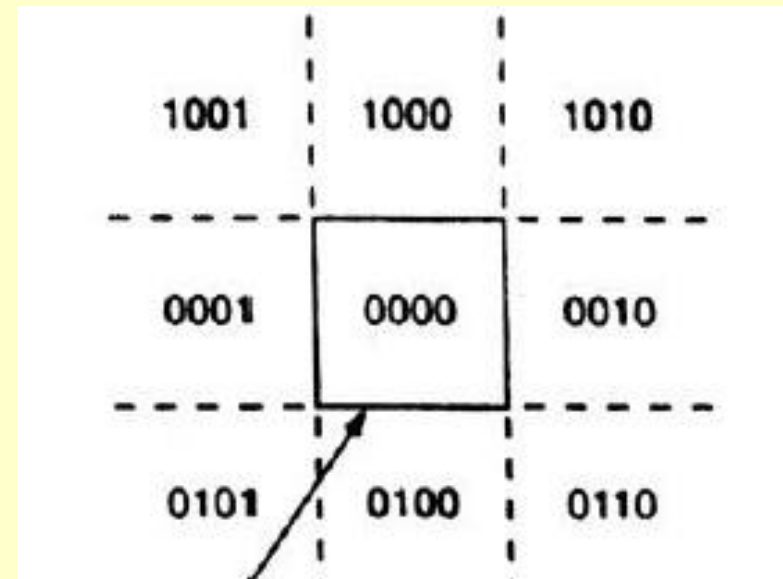
Algoritmo de Cohen-Sutherland

- Para executar os testes de aceitação ou rejeição trivial, as arestas do retângulo de visualização são estendidas de forma a dividir o plano do retângulo de visualização em 9 regiões.



Algoritmo de Cohen-Sutherland

- Bit 3 semiplano acima da aresta superior
 $y > y_{\max}$
- Bit 2: semiplano abaixo da aresta inferior
 $y < y_{\min}$
- Bit 1: semiplano a direita da aresta direita
 $x > x_{\max}$
- bit 0: semiplano a esquerda da aresta esquerda
 $x < x_{\min}$.



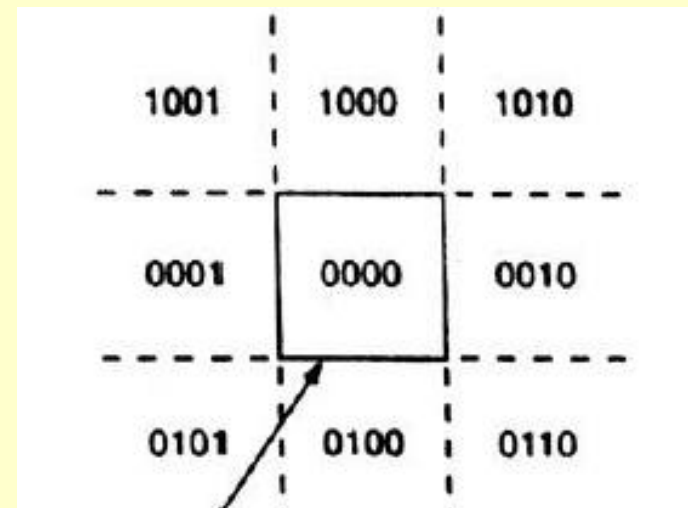
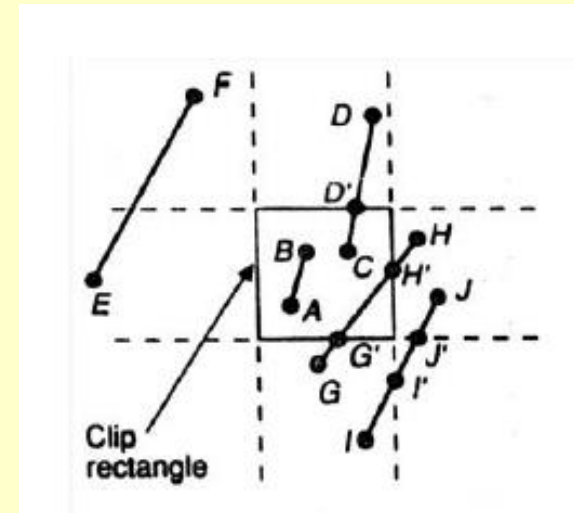
Bit3 – bit2 – bit1 – bit0

Superior – inferior – direita – esquerda

Algoritmo de Cohen-Sutherland

- A cada extremidade do segmento de reta é atribuído o código da região a qual ela pertence.
- Esses códigos são usados para determinar se o segmento está completamente dentro do retângulo de visualização ou em um semiplano externo a uma aresta.
- Se os 4 bits dos códigos das extremidades são iguais a zero, então a linha está completamente dentro do retângulo.

- Se ambas as extremidades estiverem no semi plano externo a uma aresta em particular, como para EF, os códigos de ambas as extremidades têm o bit correspondente à aresta igual a 1.
- Para EF:
- $E = 0001$ e $F = 1001$, respectivamente.
- Portanto, se um AND bit a bit dos códigos das extremidades **for diferente** de zero, o segmento pode ser trivialmente rejeitado.

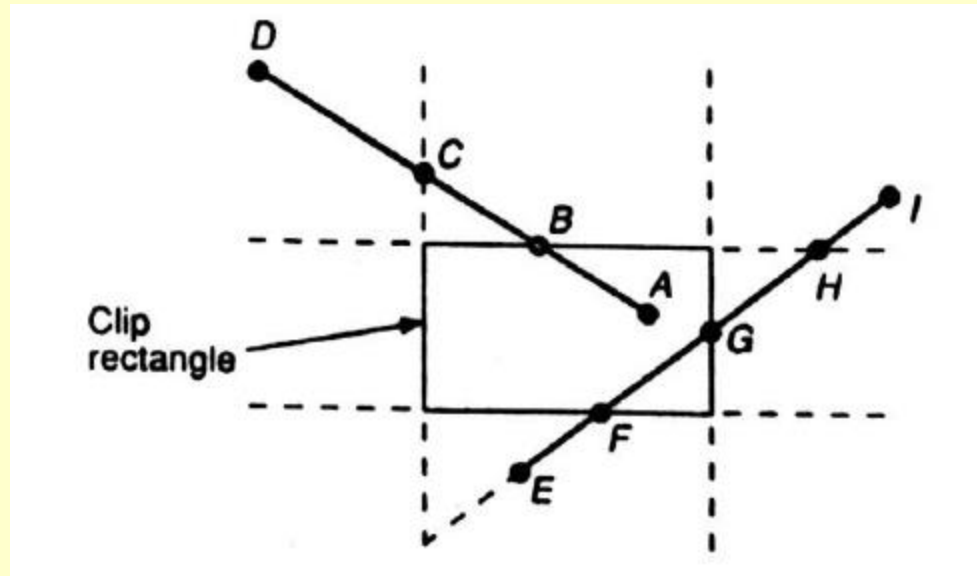


Algoritmo de Cohen-Sutherland

- Se a linha não puder ser trivialmente aceita ou rejeitada, devemos subdividi-la em dois segmentos de forma que um deles possa ser rejeitado.
- A subdivisão é feita usando uma aresta que seja interceptada pela linha para subdividi-la em dois segmentos: a seção que fica no semiplano externo à aresta é descartada.
- O teste de quais arestas interceptam a linha segue a mesma ordem dos bits nos códigos: de cima-para-baixo, da direita-para-a-esquerda.

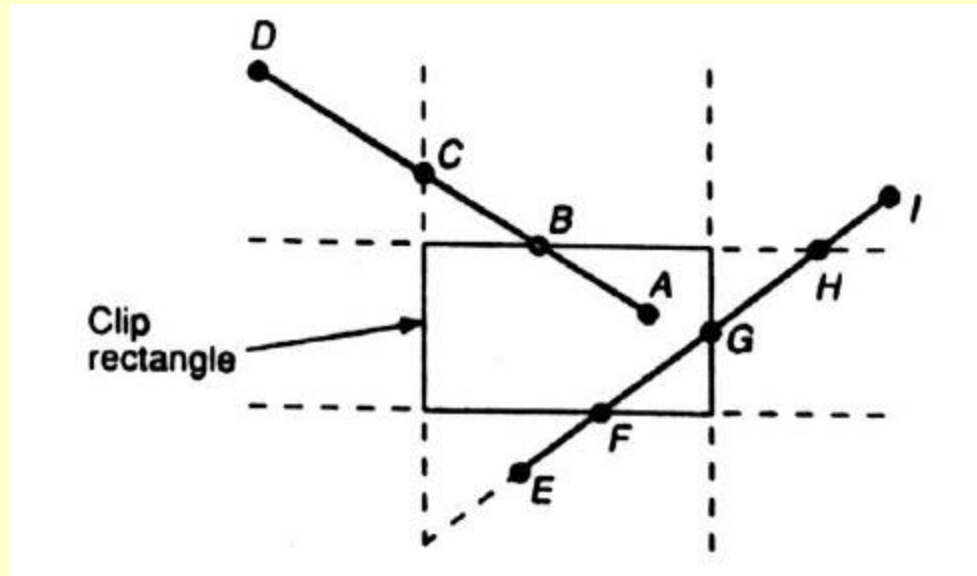
Algoritmo de Cohen-Sutherland

- Uma propriedade chave dos códigos é que os bits iguais a um correspondem a arestas interceptadas: se uma extremidade está no semiplano externo a uma aresta, e o segmento falha nos testes de rejeição trivial, então a outra extremidade deve estar no semiplano interno àquela aresta, que é portanto interceptada pelo segmento.
- O algoritmo sempre escolhe um ponto que esteja fora, e usa os bits do código iguais a um para determinar uma aresta de recorte.
- A aresta escolhida é a primeira encontrada na ordem pré-definida, ou seja, a correspondente ao primeiro bit igual a 1 no código.

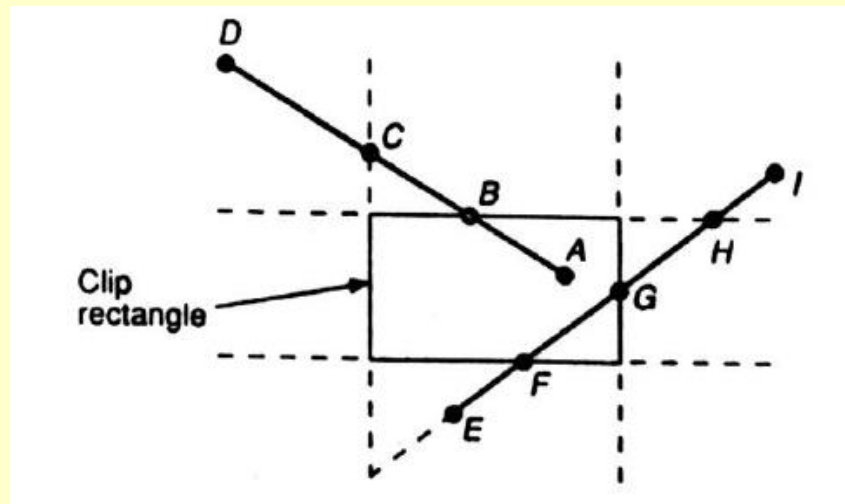


Exemplo: Segmento AD.

- Ponto A :0000 Ponto D: 1001 .
- O segmento não pode ser trivialmente aceito ou rejeitado.
- O algoritmo escolhe D como o ponto externo.
- O seu código mostra que a linha cruza as arestas superior e esquerda.

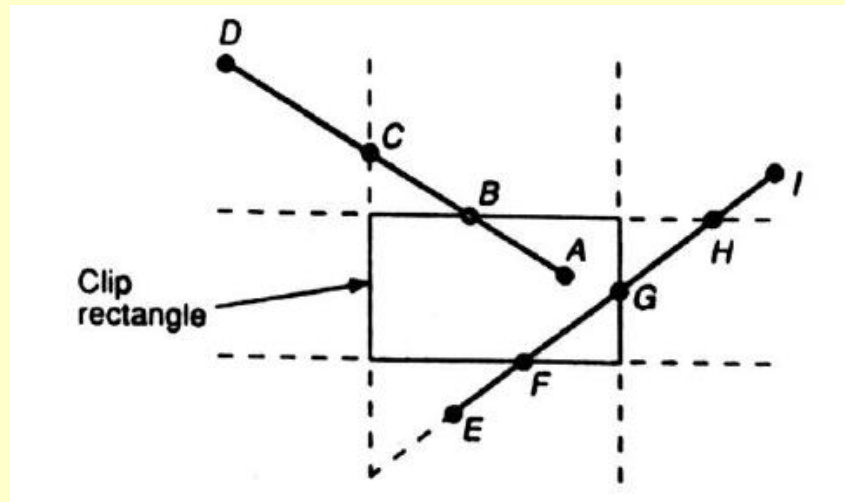


- Ordem do teste: usa-se primeiro a aresta superior para recortar AD, gerando B.
- O código de B é 0000.
- Na próxima iteração o segmento será trivialmente aceito e visualizado.



Linha EI :

- **E = 0100 I = 1010.**
- E é escolhido pelo algoritmo
- Aresta inferior gera ponto $F = 0000$
- Outra iteração: segmento não é aceito ou rejeitado trivialmente.



- **F = 0000 I = 1010.**
- I é escolhido pelo algoritmo
- Aresta superior gera ponto H = 0010
- Outra iteração, pois o segmento não é aceito ou rejeitado trivialmente.
- Na terceira iteração, corte pela aresta direita, gerando ponto **G**
- **FG é trivialmente aceito na iteração seguinte.**

Algoritmo de Cohen-Shutterland

- Não é o algoritmo mais eficiente : testes e recorte são executados numa ordem fixa, podem ser realizados cortes desnecessários. Por exemplo, quando a intersecção com uma aresta:
- Exemplo: H está em uma “intersecção externa”
- Existem algoritmos mais eficientes, mas este é muito utilizado devido à sua simplicidade.
- **Recorta a primitiva antes da conversão matricial, calculando suas intersecções com o retângulo de recorte/ visualização[clip rectangle]. Estes pontos de intersecção são então usados para definir os novos vértices para a versão recortada da primitiva.**

Clipping

- Outra estratégia seria converter todo o polígono, mas traçar apenas os pixels visíveis no retângulo de visualização. Isto pode ser feito verificando as coordenadas de cada pixel a ser escrito contra os limites do retângulo. Na prática, existem maneiras de acelerar o processo que evitam o teste de cada pixel individualmente.
- Se o teste dos limites puder ser feito rapidamente (por hardware especializado, por exemplo), esta abordagem pode ser mais eficiente que a anterior, e tem a vantagem de ser extensível a regiões de recorte arbitrárias.

Exercício

- *Descrever as iterações associadas ao algoritmo de Algoritmo de Cohen-Shutterland para o segmento AB, abaixo:*

