

## Capítulo 7

# Recorte

Uma vez gerada a imagem, ela é enviada a um dispositivo de saída para ser visualizada. Independentemente do tipo de imagem (armada, monocromática ou colorida), o dispositivo de saída deve ter a capacidade de selecionar porções de interesse que devem ser exibidas. O processo de extração da uma sub-região de interesse de uma imagem é conhecida em Computação Gráfica por **recorte** ou *clipping*. Como este processo é executado toda vez que uma cena é criada ou modificada, é desejável que ele seja o mais rápido possível para que a atualização possa ocorrer dentro da faixa de resolução temporal da visão humana (a média requerida é uma taxa de regeneração de 80 a 90 Hz).

Um recorte pode ser feito no espaço contínuo ou no espaço  $\mathbb{R}^2$  discreto, depois de amostrar a imagem. A princípio, pode-se reduzir analiticamente o problema de recorte em espaço contínuo num problema de interseção entre uma cena modelada e as faces do volume de interesse ou as bordas da janela de interesse. Sendo a maioria dos algoritmos de interseção temporalmente complexa, somente os pontos, os segmentos e figuras poligonais são tratados de forma eficiente. Para figuras geométricas mais complexas, é comum fazer o recorte depois da amostragem (Capítulo 8) e a técnica é conhecida por **tesourada** ou *scissoring*, a qual consiste em simplesmente eliminar as amostras  $(u, v)$  da imagem rasterizada que não tiverem as coordenadas contidas na janela de interesse, ou seja  $u_{min} \leq u \leq u_{max}$  e  $v_{min} \leq v \leq v_{max}$ , onde  $(u_{min}, v_{min})$  e  $(u_{max}, v_{max})$  correspondem ao canto esquerdo inferior e ao canto direito superior da janela, respectivamente.

Neste capítulo são apresentados os algoritmos analíticos de recorte básicos, que são integrados na maioria dos pacotes gráficos. Em muitas placas de vídeo, tais algoritmos são implementados em *hardware* ou *firmware* o que

melhora ainda mais a resposta do sistema. Veremos que sendo a interseção um processo computacionalmente custoso, os esforços tem sido essencialmente em reduzir o tamanho dos elementos que passam pelo algoritmo de interseção e otimizar os algoritmos de interseção.

## 7.1 Recorte de Pontos

O recorte de um ponto  $(x, y)$  em relação a uma região retangular  $(x_{min}, y_{min}, x_{max}, y_{max})$ , orientada em relação aos eixos, consiste essencialmente na classificação de pertinência ou na validação do seguinte sistema de inequações:

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

Se for um ponto  $(x, y, z)$  em relação a um volume paralelepipedal, acrescenta-se mais uma inequação ao sistema:

$$z_{min} \leq z \leq z_{max}$$

## 7.2 Recorte de Segmentos

O recorte de um segmento em relação a uma janela de interesse pode ser processado em dois passos:

1. particionamento do segmento em sub-segmentos trivialmente classificáveis (ou totalmente contidos ou totalmente externos à janela de interesse) e
2. classificação de cada subsegmento.

(Ver Fig. 3.38 do livro-texto de Foley.)

Para classificação dos segmentos ou sub-segmentos, o algoritmo mais eficiente é o algoritmo **Cohen-Sutherland** que consegue distinguir os segmentos dentro e uma grande parte dos segmentos fora de uma janela de interesse retangular definida pelos pontos  $(x_{min}, y_{min})$  e  $(x_{max}, y_{max})$ . Este algoritmo consiste em dividir um plano em 9 regiões e atribuir a cada região um código de quatro bits seguindo a seguinte convenção, do bit menos significativo para o mais:

- primeiro bit:  $x < x_{min}$ .
- segundo bit:  $x > x_{max}$ .

- terceiro bit:  $y < y_{min}$ .
- quarto bit:  $y > y_{max}$ .

e classificar os extremos de um segmento de acordo com estes códigos. Se os dois códigos forem iguais a 0000, então o segmento está dentro da janela de interesse. Senão aplica-se a operação AND, *bit a bit*, entre os dois códigos. Se o resultado for diferente de 0000, então o segmento está trivialmente fora da janela de interesse.

(Ver Fig. 3.39 do livro-texto de Foley.)

**Exercício 7.1** *Dada uma janela definida pelos pontos  $(-3, 1)$  e  $(2, 6)$ . Aplique o algoritmo de Cohen-Sutherland para separar os seguintes segmentos que são trivialmente dentro e fora da janela:  $\{(-2, 10), (-4, 7)\}$ ,  $\{(-1, 7), (-4, 2)\}$ ,  $\{(-2, 3), (1, 2)\}$ ,  $\{(3, 3), (1, -2)\}$ ,  $\{(-1, 5), (3, 8)\}$  e  $\{(4, 8), (-5, 1)\}$ .*

Quando um segmento não for trivialmente classificável, isto é o resultado da operação AND for igual a 0000, devemos particioná-lo sucessivamente em subsegmentos cujos extremos são pontos de interseção com as arestas da janela de interesse. As arestas a serem testadas podem ser facilmente identificadas pelos códigos dos pontos extremos.

(Ver Fig. 3.40 do livro-texto de Foley.)

Em 1978, Cyrus e Beck publicaram um algoritmo de recorte de segmentos em relação a qualquer janela convexa de  $n$  arestas, usando o fato de que qualquer vetor definido com os pontos sobre uma aresta  $E_i$  é perpendicular ao “vetor normal”  $N_i$  desta aresta. Define-se como vetor normal da aresta o vetor perpendicular à aresta apontando para o lado externo da janela.

(Ver Fig. 3.42 do livro-texto de Foley.)

Representando o segmento  $P_0P_1$  a ser recortado parametricamente

$$P(t) = P_0 + (P_1 - P_0)t \quad (7.1)$$

e tomando um ponto fixo  $P_{E_i}$  da aresta  $E_i$ , o ponto de interseção de  $P(t)$  com  $E_i$  deve satisfazer

$$N_i \cdot (P(t) - P_{E_i}) = 0.$$

Substituindo  $P(t)$  pela Eq.(7.1), obtemos

$$N_i \cdot (P_0 - P_{E_i}) + N_i \cdot (P_1 - P_0)t = 0$$

do qual derivamos o valor do parâmetro  $t_i$ , se o denominador for diferente de zero

$$t_i = \frac{N_i \cdot (P_0 - P_{E_i})}{-N_i \cdot (P_1 - P_0)}.$$

Se  $t_i \in [0, 1]$ , então o segmento intersecta com a aresta  $E_i$ . O ponto de interseção é classificado de acordo com o ângulo do segmento em relação ao vetor normal  $N_i$  na seguinte forma: PL (sair da janela) se o ângulo for menor que  $90^\circ$  e PE (entrar na janela) se o ângulo for maior que  $90^\circ$ . O processo é repetido para todas as arestas a fim de determinar todas as possíveis interseções. Estas interseções são então ordenadas na ordem crescente do parâmetro  $t$ . O trecho entre as interseções na sequência PE e PL é o trecho contido na janela de interesse.

(Ver Fig. 3.43 do livro-texto de Foley.)

**Exercício 7.2** *Utilize o algoritmo de Cyrus-Beck para recortar os segmentos dados no Exercício 7.1.*

**Exercício 7.3** *Liang e Barsky apresentaram uma versão mais eficiente do algoritmo de Cyrus-Beck para casos específicos de janelas retangulares. Mostre que de fato para regiões retangulares com arestas paralelas aos eixos, o algoritmo de interseção pode ser mais simples.*

### 7.3 Recorte de Polígonos

Diferentemente do recorte de segmentos, é necessário manter a conectividade das arestas de um polígono recortado para que se possa hachurar corretamente o seu interior. Isso pode ser garantido se ficarmos atentos na orientação destas arestas em relação ao polígono. Usualmente, convencionou-se que as arestas do contorno externo de um polígono são orientadas no sentido anti-horário e as arestas do contorno interno orientadas no sentido horário.

(Ver Fig. 3.46 do livro-texto de Foley.)

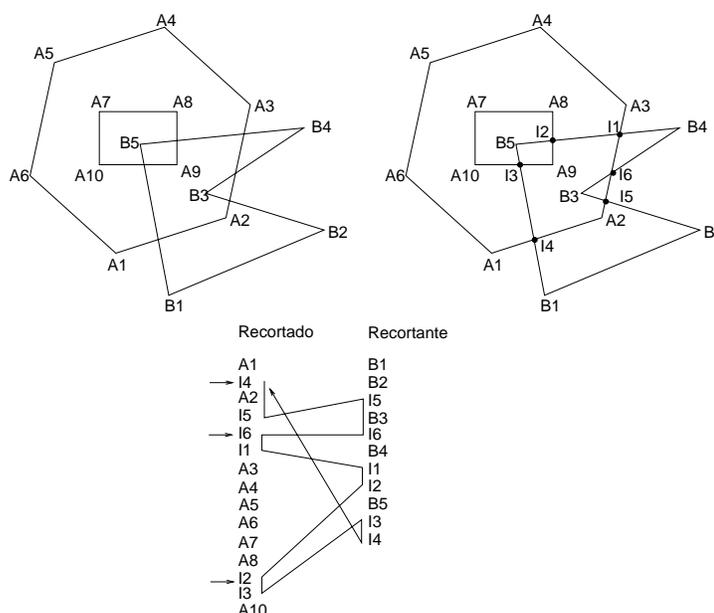
Dois algoritmos mais conhecidos para recorte de polígonos são:

1. Algoritmo de Sutherland-Hodgman: recortar recursivamente o polígono em relação a cada aresta da janela de interesse. Para cada aresta da janela, a sequência de vértices de entrada do polígono corrente é percorrida no sentido anti-horário. Enquanto os vértices estiverem no interior da janela, eles são colocados na lista de saída de vértices até que dois vértices adjacentes cruzem a aresta. Neste caso, somente um vértice e o ponto de interseção são colocados na lista de saída. O traço continua sem alterar o conteúdo da lista de saída até que a aresta seja encontrada novamente (ou seja, até que entre novamente no interior da janela), quando o ponto de interseção e os vértices do polígono passam a ser novamente colocados na lista de saída. E o processo segue até

varrer todos os vértices e a lista de saída é utilizada como o polígono de entrada para o próximo estágio de recursão.

(Ver Fig. 3.47 e 3.48 do livro-texto de Foley.)

2. Algoritmo de Weiler-Atherton: é um algoritmo bastante genérico capaz de determinar o recorte entre dois polígonos  $\mathcal{P}$  e  $\mathcal{Q}$  quaisquer. Para isso, ele define duas sequências de vértices incluindo as interseções: uma para  $\mathcal{P}$  e a outra para  $\mathcal{Q}$ . Utilizando o fato de que ao cruzar o bordo, deve-se manter ao longo do bordo até que entre novamente para o interior, o algoritmo de Weiler-Atherton percorre alternadamente as duas sequências para obter resultados de um recorte. O percurso se inicia com os pontos interseção classificados como PE em relação ao objeto recortado (na figura abaixo estes pontos são indicados por setas).



**Exercício 7.4** Considere a mesma janela definida no Exercício 7.1 e um polígono definido pela sequência externa de pontos  $\{(2,0), (0,3), (4,1), (3,8), (-4,2)\}$  e pela sequência interna de pontos  $\{(0,4), (0,5), (1,4)\}$ . Utilize o algoritmo de Weiler-Atherton para obter a porção visível do polígono.