

## Capítulo 6

# Recorte

Vimos no capítulo 5 que somente os objetos contidos no volume de visão, delimitados pelos planos  $u = u_{min}$ ,  $u = u_{max}$ ,  $v = v_{min}$ ,  $v = v_{max}$ ,  $n = n_{min} = -F$  e  $u = -B$ , são enviados a um dispositivo de saída para serem imageados. O processo de extração da uma sub-região de interesse de uma cena é conhecida em Sistemas de Informações Gráficas por **recorte** ou *clipping*. Veremos na seção 6.1 que esta extração deve ser feita antes da “homogeneização” das coordenadas  $(X, Y, Z, W)$  para  $(x, y, z, 1)$ , a fim de evitar remoções indevidas.

A princípio, pode-se reduzir analiticamente o problema de recorte num problema de interseção entre os modelos geométricos de uma cena e os planos limitantes do volume de visão. Sendo a maioria dos algoritmos de interseção computacionalmente custoso, um dos principais objetivos dos algoritmos de recorte é reduzir o tamanho dos elementos que passam pelo algoritmo de interseção e/ou otimizar os algoritmos de interseção. Exemplos de algoritmos de recorte de pontos, segmentos e polígonos são apresentados, respectivamente, nas seções 6.2, 6.3, e 6.4.

### 6.1 Espaço de Recorte

Em coordenadas homogêneas, o volume de visão normalizado é

$$-1 \leq x = \frac{X}{W} \leq 1 \quad -1 \leq y = \frac{Y}{W} \leq 1 \quad -1 \leq z = \frac{Z}{W} \leq 0,$$

ou seja, para  $W > 0$

$$-W \leq X \leq W \quad -W \leq Y \leq W \quad -W \leq Z \leq 0$$

e para  $W < 0$

$$-W \geq X \geq W \quad -W \geq Y \geq W \quad -W \geq Z \geq 0$$

no espaço de coordenadas homogêneas.

(Ver Fig. 6.57 do livro-texto de Foley.)

Se tivermos um segmento cujos extremos  $P_1$  e  $P_2$  tenham valores  $W$  de sinais opostos, ele é, no plano  $w = 1$ , um segmento “externo” de  $P_1$  e  $P_2$  (ver seção 3.3). E se  $P_1$  e  $P_2$  estiverem dentro do volume de visão, o resultado do recorte deve ser um conjunto de 2 segmentos distintos, ao invés de um segmento totalmente contido no volume de visão.

(Ver Fig. 6.58 do livro-texto de Foley.)

Com a homogeneização os pontos do plano projetivo são levados de volta para o plano afim, que não inclui os pontos do “infinito”, não existem portanto o conceito de segmentos externos. Com isso, não se consegue fazer corretamente o recorte no caso mencionado.

## 6.2 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:

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

## 6.3 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.)

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.)

**Observação 6.1** *O algoritmo de Cohen-Sutherland pode ser facilmente estendido para um volume de interesse, se adicionarmos mais 2 bits ao código: quinto bit para  $z < z_{min}$  e sexto bit para  $z > z_{max}$ .*

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

$$P(t) = P_0 + (P_1 - P_0)t \quad (6.1)$$

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.(6.1), 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  $N_i$ , ou seja  $-N_i \cdot (P_1 - P_0)$ , 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.)

Liang e Barsky apresentaram uma versão mais eficiente do algoritmo de Cyrus-Beck para casos específicos de janelas retangulares, ou seja,

- $x_{min}$ , cuja normal é  $(-1, 0)$  e  $t_i = \frac{-(x_0 - x_{min})}{x_1 - x_0}$ ;
- $x_{max}$ , cuja normal é  $(1, 0)$  e  $t_i = \frac{x_0 - x_{max}}{-(x_1 - x_0)}$ ;
- $y_{min}$ , cuja normal é  $(0, -1)$  e  $t_i = \frac{-(y_0 - y_{min})}{y_1 - y_0}$ ;
- $y_{max}$ , cuja normal é  $(0, 1)$  e  $t_i = \frac{y_0 - y_{max}}{-(y_1 - y_0)}$ ;

Para checar se o segmento está saindo ou entrando na janela, basta testar a diferença entre as coordenadas  $x$  ou  $y$  dos pontos extremos do segmento.

## 6.4 Recorte de Polígonos

Diferentemente do recorte de segmentos, é necessário manter a conectividade das arestas de um polígono recortado para que se possa “hachurar/identificar” 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

sejam 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:

**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.)

**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 a fronteira, deve-se manter ao longo da fronteira 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).

