

Computação Gráfica

Clipping em 3D

Sumário

- Volume de Clipping
- Método para rotular regiões
- Clipping ou recorte em 3 dimensões
 - Clipping de ponto
 - Clipping de segmentos de reta
 - Clipping de polígonos

Introdução

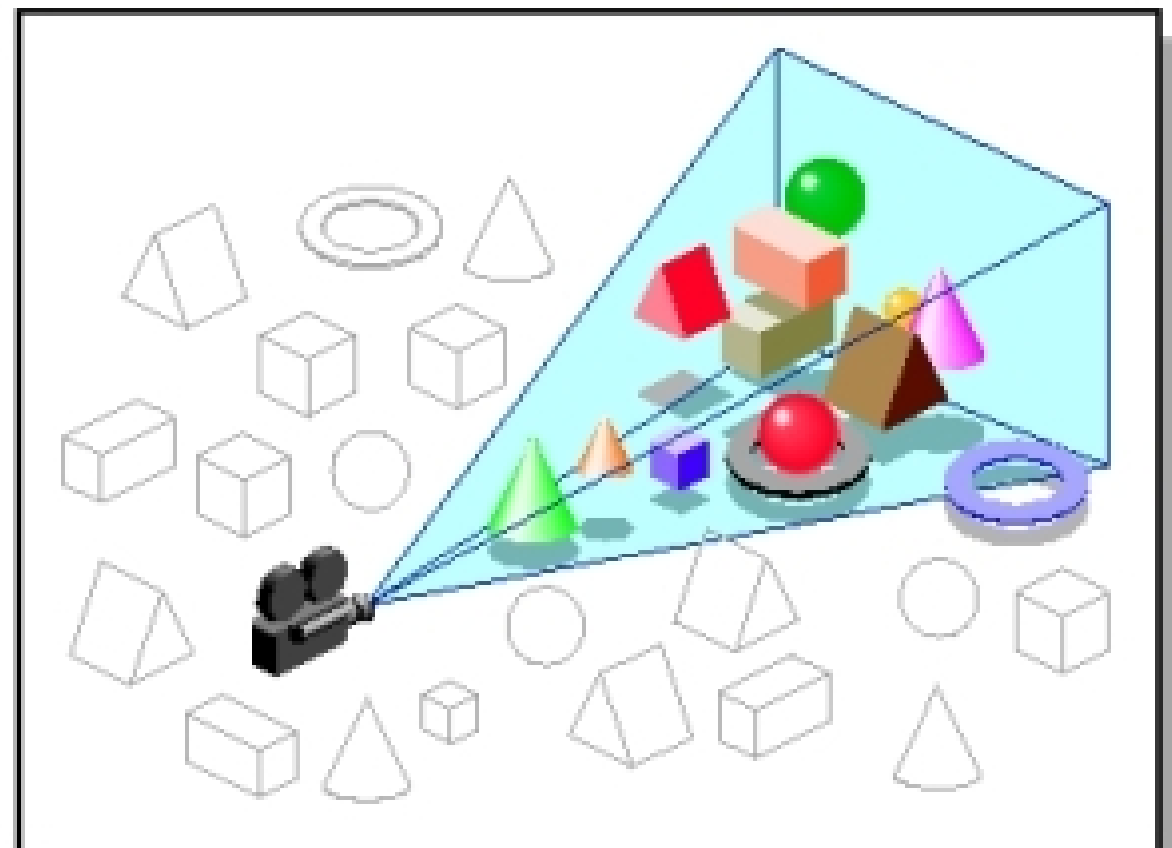
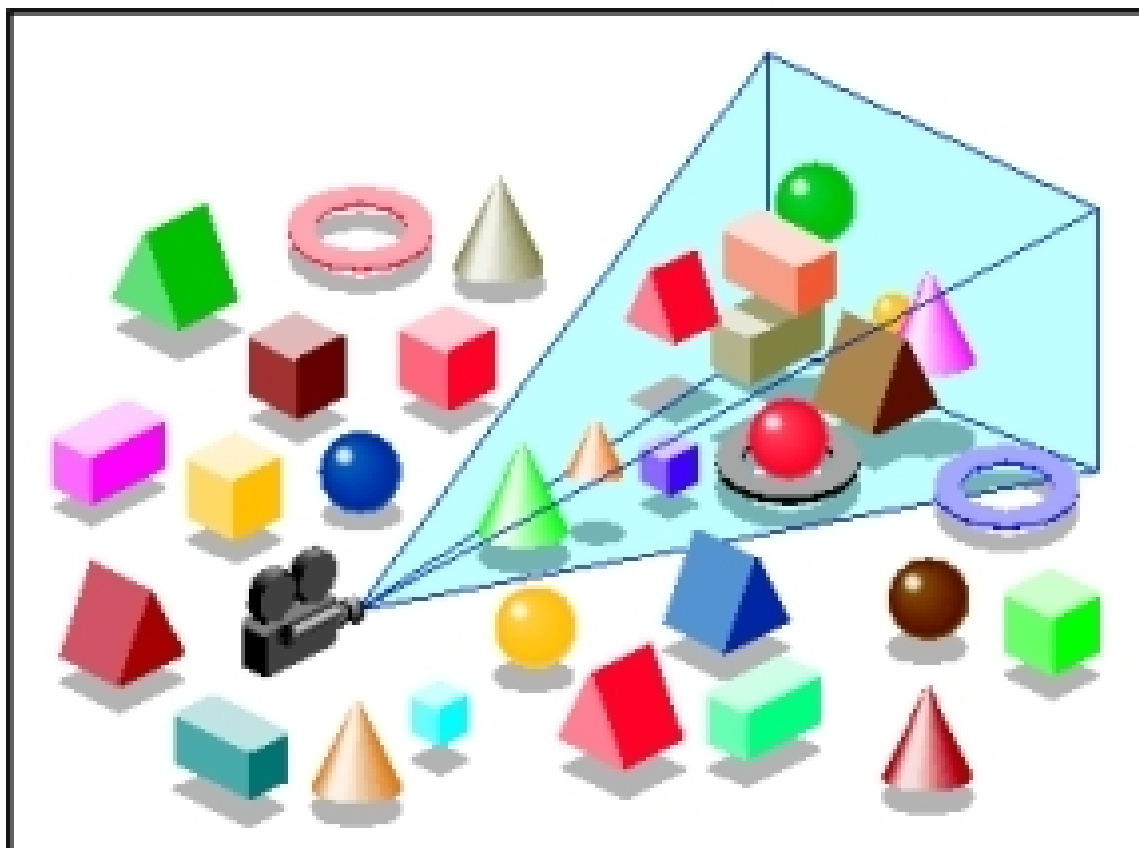
- Assim como realizado em duas dimensões, o recorte 3D tem por objetivo remover objetos que não serão visíveis a partir da cena renderizada.
 - algoritmos de clipping em três dimensões identificam e armazenam todos os objetos dentro do volume de visualização para exibi-los em dispositivos de saída.
 - objetos ou partes de objetos que se encontram fora do volume de visualização são eliminados.
- Novamente, o objetivo principal é a redução do esforço computacional.

Introdução

- O recorte em três dimensões é obtido em duas etapas:
 1. Descartar objetos que não estão visíveis
 - isto é, objetos que estão atrás da câmera, fora do campo de visão ou muito distantes.
 2. Recortar objetos que interceptam algum plano de corte (clipping plane).

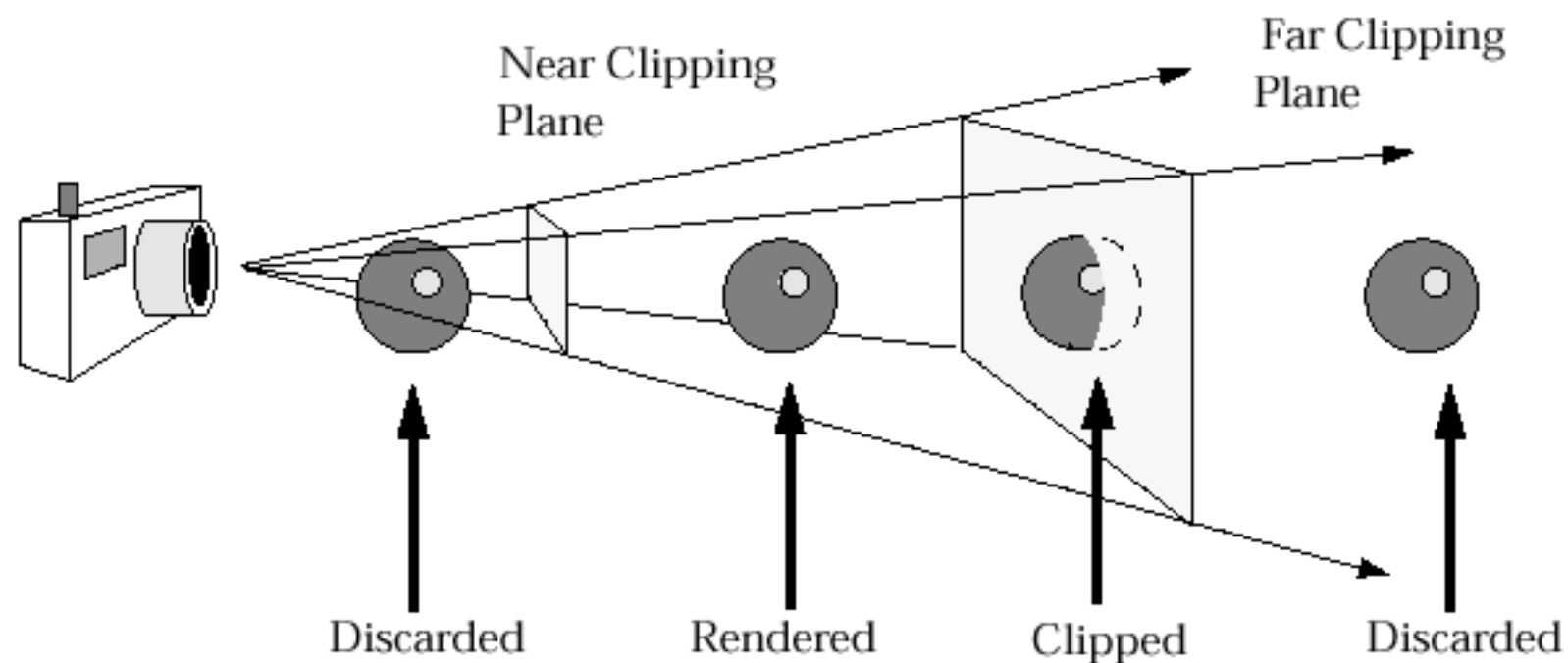
Descartando objetos

- Descartar objetos que não podem ser vistos envolve a comparação de *bounding box* ou esferas que envolvem estes objetos com relação ao volume de visualização.
- este procedimento pode ser realizado antes ou depois das transformações de projeção.



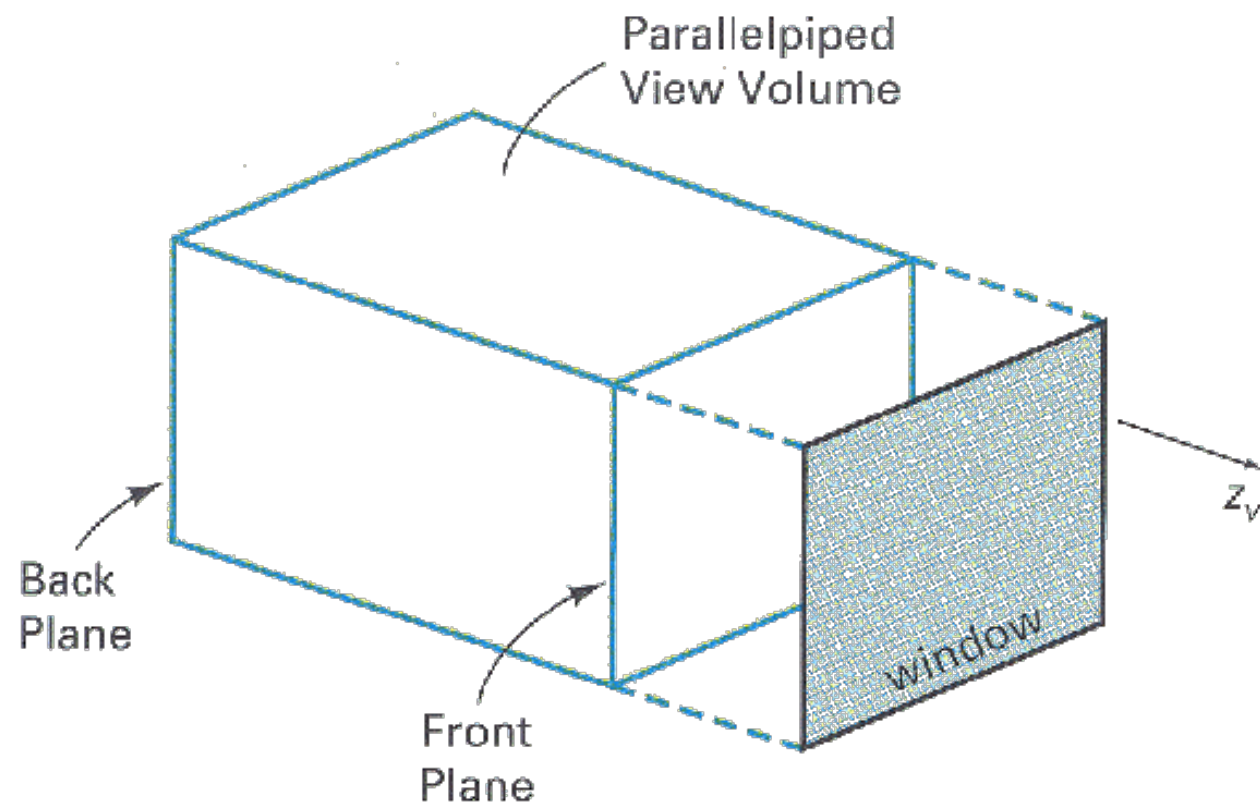
Recortando Objetos

- Objetos que estejam parcialmente inseridos no volume de visualização devem ser recortados, assim como acontece em duas dimensões.

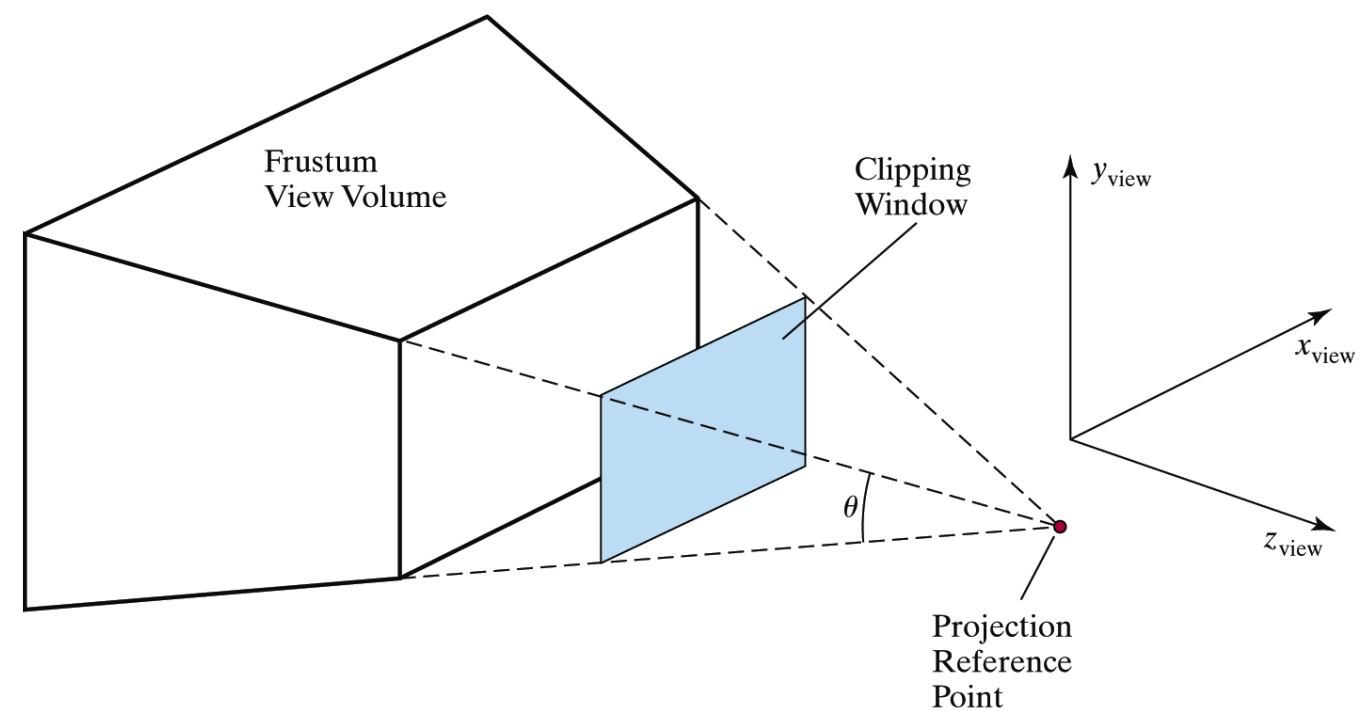


Volume de Recorte

- Transformação de Projeção Ortogonal



- Transformação de Projeção Perspectiva



Volume de Recorte

- Terminada a transformação de projeção perspectiva, o volume de visualização (frustum) é convertido em um paralelepípedo.

Preservadas

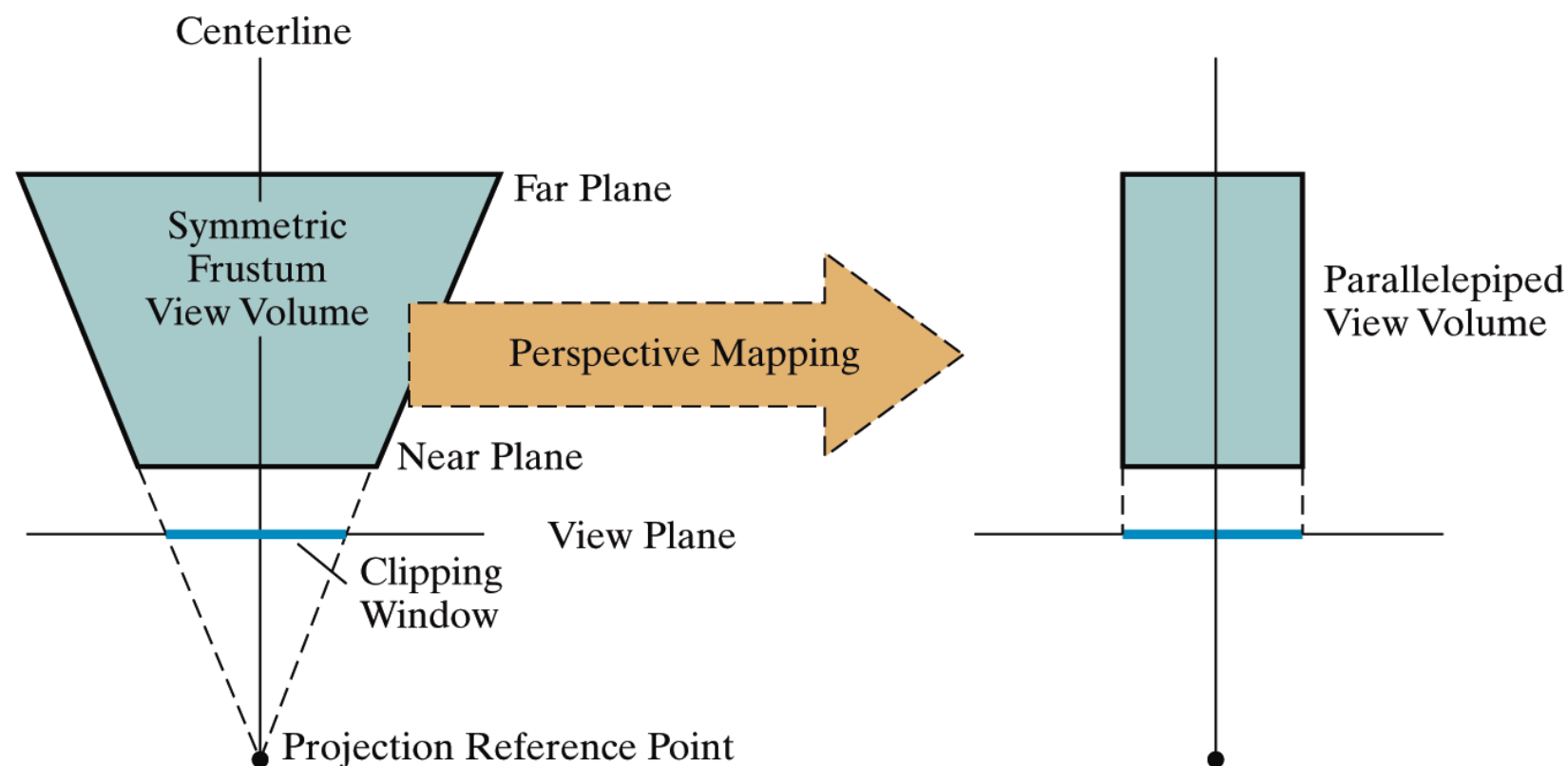


Figure 7-51

A symmetric frustum view volume is mapped to an orthogonal parallelepiped by a perspective-projection transformation.

Normalização

- Os volumes de visualização são normalizados com relação à posição $(-1, -1, -1)$ e o eixo z é revertido.

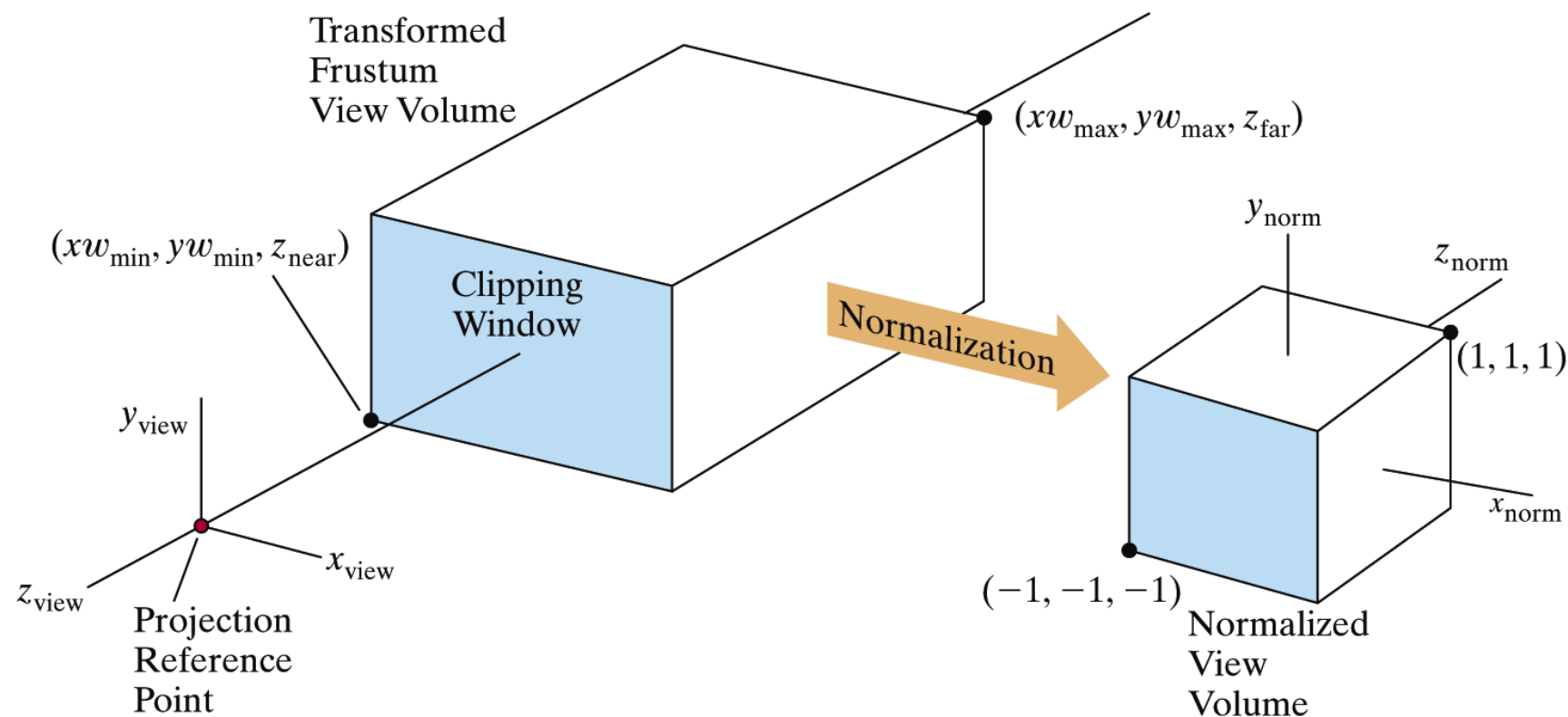


Figure 7-53

Normalization transformation from a transformed perspective-projection view volume (rectangular parallelepiped) to the symmetric normalization cube within a left-handed reference frame, with the near clipping plane as the view plane and the projection reference point at the viewing-coordinate origin.

O que é recortado?

- O recorte é realizado após a transformação de projeção e posterior normalização terem sido concluídas.

- Então, temos o seguinte:

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = M \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- onde ***M*** representa a concatenação das várias transformações aplicadas desde a representação em coordenadas do mundo até a normalização.
- O recorte é aplicado sobre estas coordenadas homogêneas.

Divisão do Espaço 3D em Regiões

- Para o *clipping* 3D, o conceito de códigos de região visto anteriormente é estendido por meio de dois *bits* adicionais na codificação das regiões.
- estes *bits* representam os planos de recorte (clipping planes) frontal e Posterior.
- novamente, 0 = dentro do volume de recorte,
1 = fora do volume de recorte
- os códigos em cada região são representados como a seguir:

bit 6	bit 5	bit 4	bit 3	bit 2	bit 1
Posterior	Frontal	Acima	Abaixo	Direito	Esquerdo

Divisão do Espaço 3D em Regiões

- Uma vez que o volume de recorte é normalizado, um ponto se encontra dentro do volume se suas coordenadas de projeção satisfazem às seguintes equações:

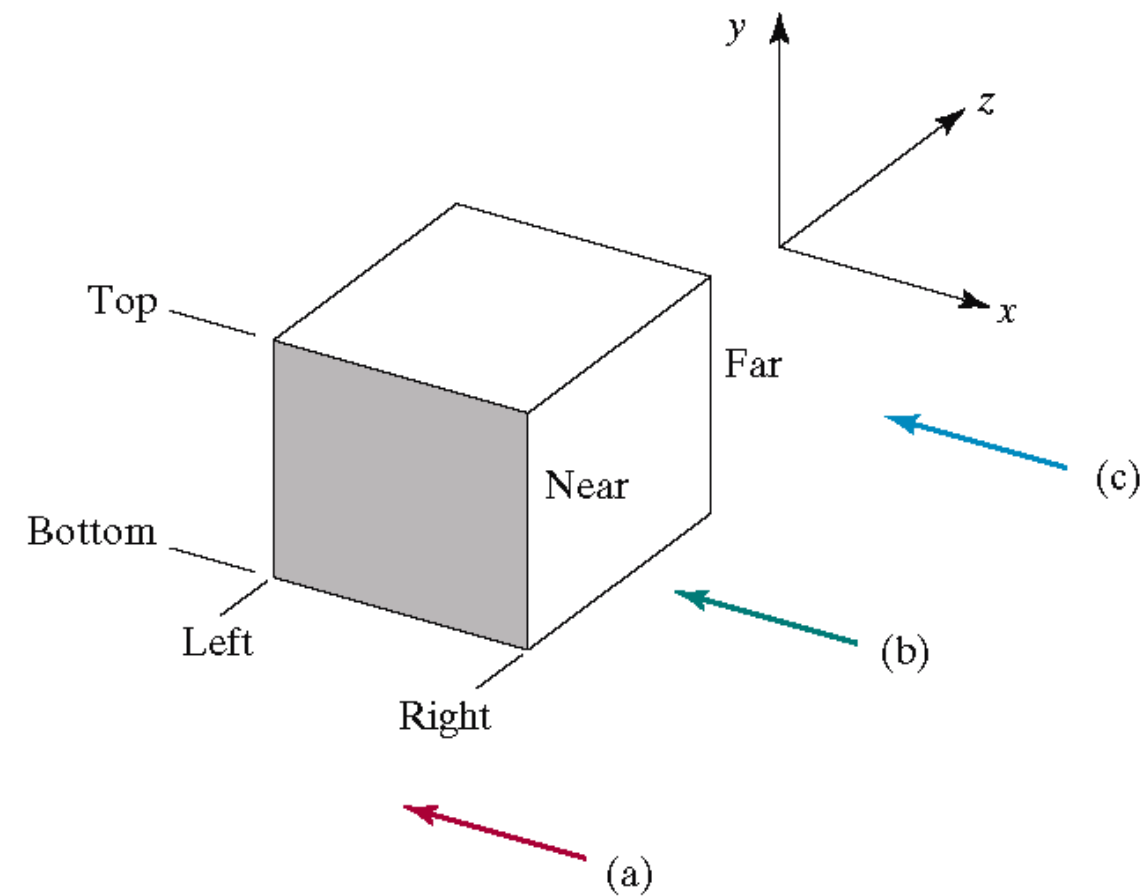
$$-1 \leq \frac{x_h}{h} \leq 1 \quad -1 \leq \frac{y_h}{h} \leq 1 \quad -1 \leq \frac{z_h}{h} \leq 1$$

- lembre-se que cada ponto na cena é representado por $P = (x_h, y_h, z_h, h)$
- Reajustando as equações acima temos:

$$-h \leq x_h \leq h \quad -h \leq y_h \leq h \quad -h \leq z_h \leq h \quad \text{se } h > 0$$

$$h \leq x_h \leq -h \quad h \leq y_h \leq -h \quad h \leq z_h \leq -h \quad \text{se } h < 0$$

Códigos de Região 3D



011001	011000	011010
010001	010000	010010
010101	010100	010110

Region Codes
In Front of Near Plane
(a)

001001	001000	001010
000001	000000	000010
000101	000100	000110

Region Codes
Between Near and Far Planes
(b)

101001	101000	101010
100001	100000	100010
100101	100100	100110

Region Codes
Behind Far Plane
(c)

Point Clipping

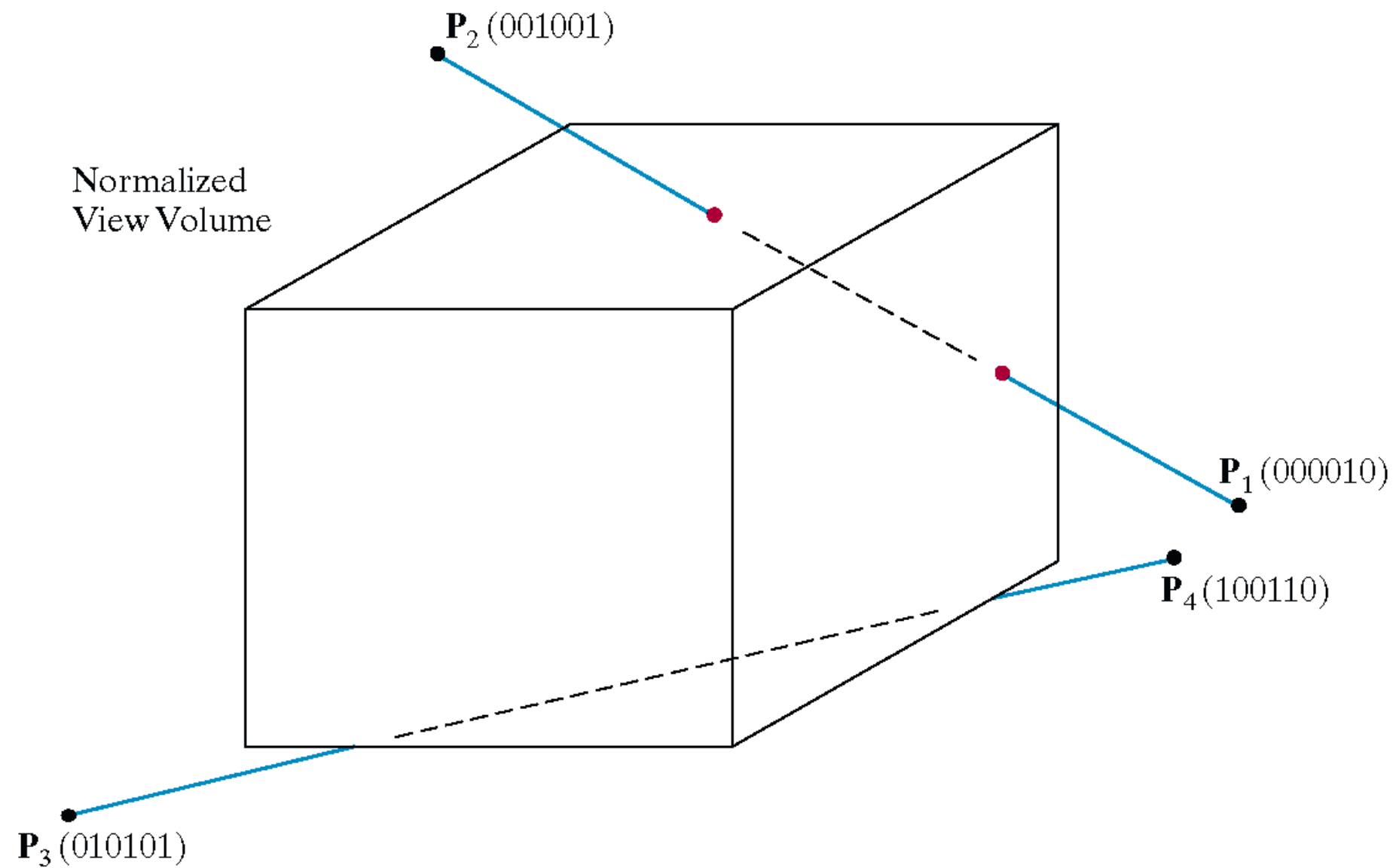
- O recorte de pontos em três dimensões é trivial.
 - todo ponto que não possua código de região igual a [000000] é eliminado.

Line Clipping

- Métodos para o recorte de segmentos de reta em três dimensões são essencialmente iguais àqueles aplicados em duas dimensões.
 - o primeiro passo é rotular todos os pontos extremos de acordo com a região apropriada.
 - segmentos de reta cujos pontos extremos possuam, ambos, códigos iguais a [000000] são mantidos para a renderização.
 - segmentos de reta cujos pontos extremos compartilham um bit em qualquer posição podem ser recortadas (igual ao caso bidimensional).

Line Clipping

Linha P_3P_4 pode ser totalmente eliminada.



Equação da Reta para Clipping 3D

- Equações de recorte para segmentos de reta tridimensionais são descritas em termos de sua representação paramétrica.
- Para um segmento de linha formado pelos pontos $P_1 = (x_{1h}, y_{1h}, z_{1h}, h_1)$ e $P_2 = (x_{2h}, y_{2h}, z_{2h}, h_2)$, a equação paramétrica que descreve cada ponto deste segmento é dada por:

$$P = P_1 + (P_2 - P_1)u \qquad 0 \leq u \leq 1$$

Equação da Reta para Clipping 3D

- Representando esta equação paramétrica do segmento de reta em termos de coordenadas homogêneas temos

$$x_h = x1_h + (x2_h - x1_h)u$$

$$y_h = y1_h + (y2_h - y1_h)u$$

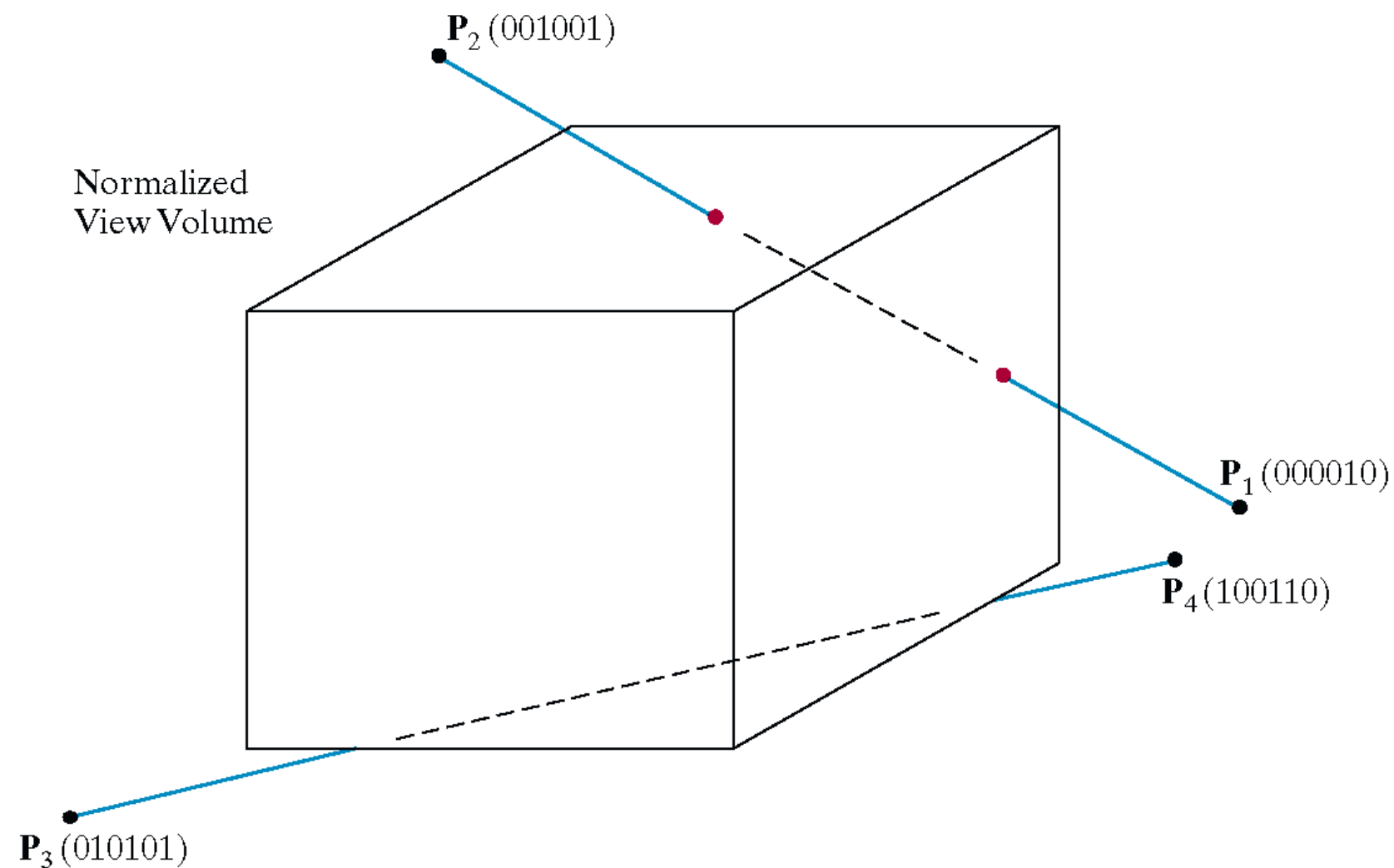
$$z_h = z1_h + (z2_h - z1_h)u$$

$$h = h1 + (h2 - h1)u$$

Line Clipping

- Considere o segmento de linha dado pelos pontos $P_1[000010]$ e $P_2[001001]$

Plano direito



Line Clipping

- Uma vez que o plano direito pode ser descrito pela equação $x = 1$, a seguinte equação é válida

$$x_p = \frac{x_h}{h} = \frac{x1_h + (x2_h - x1_h)u}{h1 + (h2 - h1)u} = 1$$

- e, resolvida com relação ao parâmetro u , temos

$$u = \frac{x1_h - h1}{(x1_h - h1) - (x2_h - h2)}$$

- utilizando este valor de u é possível obter os valores de y_p e z_p .
- A partir daí basta seguir de forma idêntica ao algoritmo de recorte de linha 2D.

Clipping de Polígonos

- O caso mais comum no recorte 3D diz respeito a objetos gráficos criados a partir de polígonos

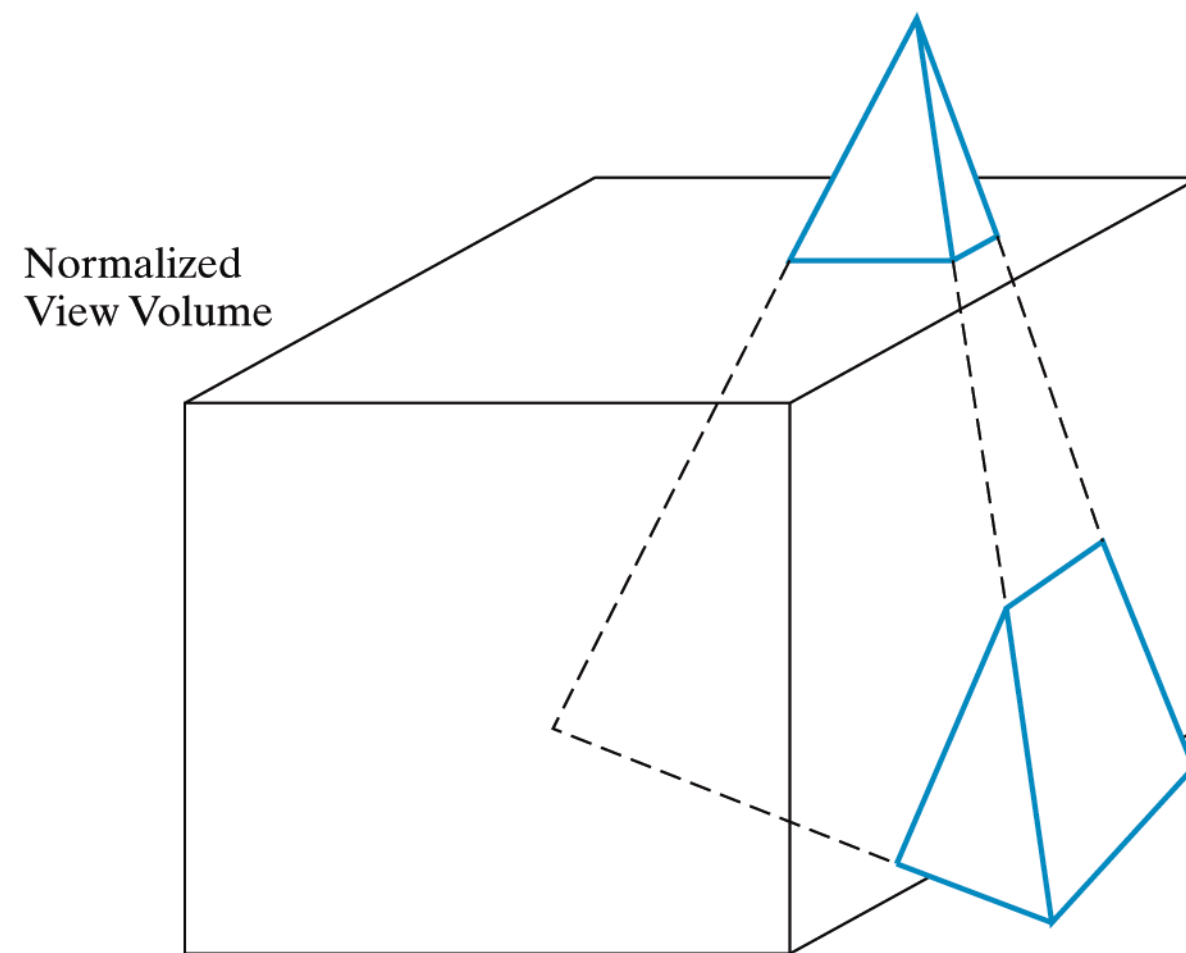


Figure 7-59

Three-dimensional object clipping. Surface sections that are outside the view-volume clipping planes are eliminated from the object description, and new surface facets may need to be constructed.

Clipping de Polígonos

- Neste caso, primeiramente, tenta-se eliminar os objetos utilizando *bounding boxes* ou esferas.
- A seguir, é realizado o recorte de polígonos individuais utilizando o algoritmo de Sutherland-Hodgman já apresentado.