

Capítulo 4

Modelagem de Cenas

Modelagem de cenas consiste, essencialmente, em agrupar os modelos geométricos para criar um cenário de interesse. Usualmente, estes modelos geométricos são definidos em separado, num sistema de referência próprio, por motivos apresentados na seção 4.1.

(Ver Fig. 5.26 do livro-texto de Foley.)

Teoricamente, o problema se reduz em redefinir um sistema de referência comum e posicionar todos os objetos neste novo sistema. Uma consequência prática da facilidade de agrupar uma variedade de modelos numa mesma cena é que o tempo de imageamento de uma cena contendo milhares de facetas pode tornar crítico. Isso motivou trabalhos dirigidos para estruturação dos elementos da cena, de forma a permitir eliminação eficiente de facetas não visíveis.

No capítulo 3 apresentamos transformações mais usuais num mesmo sistema de referência. Mostramos ainda que utilizando representações invariantes sob transformações afins, podemos melhorar o desempenho do sistema aplicando-as somente sobre os vértices das malhas poligonais. Na seção 4.2 investigaremos as modificações a serem feitas sobre as coordenadas cartesianas dos pontos dadas em termos de um sistema de referência A , quando mudamos o referencial A para um outro referencial B .

Na seção 4.3 comentaremos dois paradigmas de organização dos dados de uma cena para aumentar a eficiência no imageamento de cenas complexas.

4.1 Modelagem

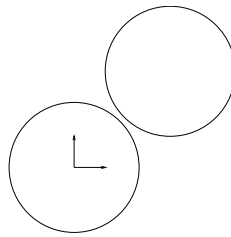
Por simplicidade de modelagem e por usabilidade, o paradigma **dividir-para-conquistar** (*divide-and-conquer*) é muito utilizado no processo de

modelagem de uma cena.

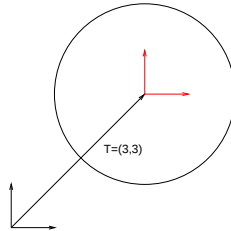
Usualmente, os objetos são modelados num sistema de referência conhecido por **sistema de objeto**. Um dos fatores que influem na escolha deste sistema é a simplicidade do modelo do objeto. Por exemplo, uma circunferência de raio igual a 2 centrada na origem e no ponto (3,3) pode ser descrita pelas seguintes equações implícitas:

1. $x^2 + y^2 = 4$, e
2. $x^2 - 6x + y^2 - 6y + 18 = 4$.

Ambas representam a mesma figura geométrica, mas a segunda envolve mais parâmetros e é menos direta para ser percebida como o lugar geométrico de uma circunferência.



Uma alternativa, utilizada no processo de modelagem de cenas, é descrever a segunda circunferência como a primeira circunferência mudando de um espaço (do objeto, em vermelho) para o outro (espaço do mundo, em preto). Com isso, centraremos nas transformações entre os espaços, ao invés de nos determos nos pontos/vértices que constituem os modelos. Neste caso particular, a transformação entre os espaços é uma translação (3,3) da origem.



Esta estratégia nos permite replicar o mesmo modelo, quantas vezes que quisermos, e aplicarmos sobre as coordenadas das suas réplicas as transformações afins para posicioná-las apropriadamente numa cena. Falta agora verificar como se pode computar, de forma eficiente, as coordenadas cartesianas dos pontos dos referenciais transformados.

4.2 Transformações entre Referenciais

Foi introduzido na seção 2.1.1 o conceito de coordenadas afins $(\alpha_1, \alpha_2, \alpha_3)$, as quais podem ser consideradas os múltiplos dos vetores-base que se combinam linearmente para gerar o vetor-posição de um ponto P . Comentamos ainda que, quando os vetores-base forem canônicos — $(1, 0, 0)$, $(0, 1, 0)$ e $(0, 0, 1)$, estas coordenadas coincidem com as coordenadas cartesianas (x_v, y_v, z_v) , que são as mais utilizadas para referenciar os pontos nos sistemas de informações gráficas, ou seja,

$$\begin{bmatrix} x_v \\ y_v \\ z_v \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

Se aplicarmos sobre os vetores-base uma transformação linear (mudança de base ou mudança de referencial) e preservarmos os modelos, ou seja, mantermos as suas coordenadas afins

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix},$$

teremos

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_v \\ y_v \\ z_v \end{bmatrix}.$$

Ou seja, as novas coordenadas cartesianas dos pontos correspondem às coordenadas cartesianas originais multiplicadas pela mesma matriz de transformação linear de espaço.

Ao transladarmos a origem O dos vetores-base do sistema de referência pelo montante

$$\begin{bmatrix} dx \\ dy \\ dz \end{bmatrix},$$

os vetores-base devem ser adicionados deste montante; portanto, as coordenadas cartesianas dos pontos passarão a ser

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} + \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} x_v \\ y_v \\ z_v \end{bmatrix} + \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix}.$$

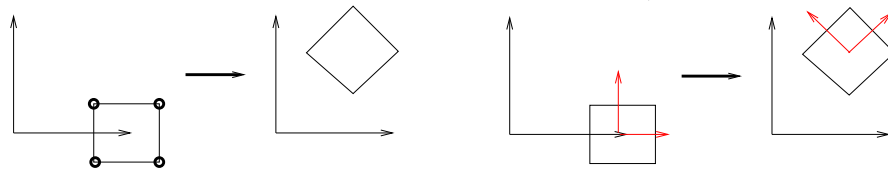
Concluindo, para uma dada transformação afim T , obteremos as mesmas coordenadas cartesianas P ,

1. quando aplicarmos a transformação diretamente sobre elas; ou
2. quando aplicarmos a transformação sobre os vetores-base B do espaço corrente e determiná-las a partir das combinações lineares destes utilizando as mesmas coordenadas cartesianas.

Em outras palavras, são equivalentes os efeitos de transformações afins T sobre as coordenadas P e os seus efeitos sobre os vetores-base do sistema de referência B , definidos por quatro pontos não coplanares,

$$B(TP) = (BT)P = (TB)P = (TB)P.$$

No seguinte exemplo, a rotação do quadrado em torno da origem do sistema referencial B (aplicar a matriz de transformação sobre as coordenadas cartesianas dos vértices) é equivalente a rodar o sistema referencial A , no qual o quadrado está centrado, em relação ao referencial B (recalcular as coordenadas cartesianas como combinações lineares dos vetores-base transformados pela mesma transformação de rotação).



Rotação dos 4 vértices em torno da origem

Rotação do sistema A em torno da origem

4.3 Estruturação

Fazendo uso da equivalência, a modelagem de cenas é comparativamente mais simples do que a modelagem de objetos. Como a complexidade de imageamento de uma cena é dependente da quantidade de facetas triangulares, uma cena constituída por muitos objetos ou por objetos muito complexos pode requerer um tempo excessivamente elevado. Para aumentar a eficiência de imageamento, existem propostas de organizar estas facetas em estruturas, a partir das quais pode-se, por exemplo,

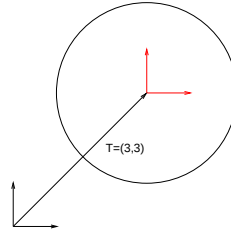
1. ajudar no descarte de superfícies não visíveis e reduzir o espaço de facetas a serem processadas,

2. ordenar facilmente as facetas em relação ao observador.

Para o primeiro caso, aplica-se o conceito de **ocupação espacial** utilizada para modelagem de objetos

(Ver Fig. 12.20 – 12.23 do livro-texto de Foley.)

para organizar as facetas num volume subdividido em *voxels* (*volume elements*), de forma que se pode descartar trivialmente os *voxels* não visíveis e limitar o espaço de facetas a serem processadas às contidas nos *voxels* visíveis.



Para o segundo caso, a árvore BSP (*binary-partitioning space*) e suas variantes são as mais conhecidas. O princípio básico da construção desta árvore é subdividir um espaço S (um nó-raiz) em dois sub-espacos, S_1 e S_2 (dois nós ligados a ele), por meio de uma faceta contida em S e classificar S_1 e S_2 como “no lado do vetor normal” da faceta e “no outro lado”, e assim sucessivamente até que nos sub-espacos só contenham no máximo uma faceta (nós terminais). Com esta árvore, pode-se ordenar a sequência das facetas em relação a um observador posicionado em qualquer lugar, desde que a percorra recursivamente coletando os nós na ordem “no lado do vetor-normal” – nó (faceta) – “no outro lado”. Voltaremos a falar sobre este tópico no Capítulo 10.

(Ver Fig. 12.26 do livro-texto de Foley.)