



UNIVERSIDADE FEDERAL DO TOCANTINS

TRABALHO AVALIATIVO DE COMPUTAÇÃO GRÁFICA

DISCENTE: Fábio Manuel Martins Tronção
DOCENTE: Eduardo Ferreira Ribeir
E-mail:fabio.troncao@mail.uft.edu.br

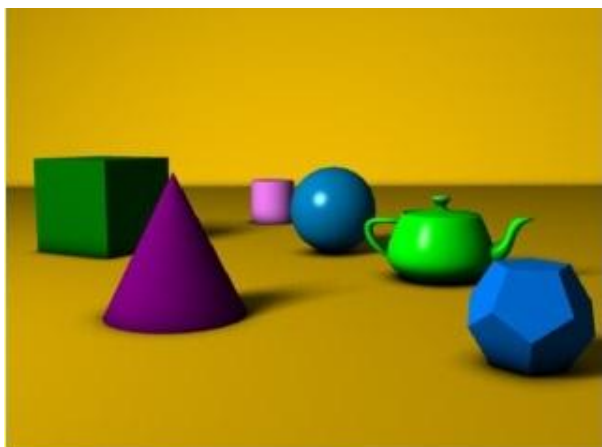
Z-buffer

Z-buffer, também chamado de buffer de profundidade (depth buffer), é uma técnica utilizando buffer de dados usada na computação gráfica que consiste na memória de um adaptador de vídeo encarregada de gerir as coordenadas de profundidade das imagens nos gráficos em três dimensões (3-D), de forma a eliminar sobreposições, normalmente calculadas por hardware e algumas vezes por software.

Para que se possa realizar a representação precisa e realista de um modelo tridimensional complexo, numa cena contendo vários objetos diferentes, é necessário que aquelas superfícies normalmente invisíveis de um determinado ângulo ou ponto no mundo sejam também renderizadas invisíveis no computador. Este problema é normalmente chamado de problema de superfície visível (visible-surface problem), quando o objetivo é determinar quais superfícies de um objecto são visíveis de um determinado ângulo, ou problema de superfície oculta (hidden-surface problem), quando o objectivo é determinar quais superfícies estão invisíveis neste momento.

Foi um problema fundamental abordado pela pesquisa em Computação Gráfica durante muito tempo. Para ilustrar o que são os efeitos desejados de um método dessa família de algoritmos, basta observar os dois desenhos mostrando a mesma estrutura, uma em modelo de arame e outra em modelo de sólidos/superfícies. A representação com z-buffer aplicado fornece de imediato muito mais informação sobre as posições relativas dos objetos na cena, uma vez que não só acontece a oclusão por sobreposição de pixels, mas também de intensidade dos pixels, ambas as coisas utilizando a coordenada z.

De todos os algoritmos para determinação de superfície visível, o buffer de profundidade ou Z-buffer é talvez o método mais simples e com certeza o mais amplamente utilizado.



Uma cena tridimensional simples



Representação Z-buffer

O princípio de funcionamento do algoritmo é muito simples: Para todos os pixels na Viewport, mantêm-se um registo da profundidade (em termos de coordenada z) do objeto na cena em que estiver mais próximo, para além de um registo da intensidade, cor, etc, que deveria ser utilizada ao mostrar esse ponto em particular na tela do computador. Todas as vezes que um novo polígono é processado, um valor de z , e de intensidade são calculados para cada pixel que estiver dentro dos limites do polígono. Se o valor de coordenada z obtido para aquele polígono for inferior ao valor de z armazenado para aquele pixel no buffer, então esse objeto está mais próximo do que algum objeto anteriormente renderizado naquela posição e vai ocultar aquele objeto. Neste caso, substitui-se o valor armazenado naquela posição do buffer por este novo valor.

Se não for realizado nenhum cálculo de radiosidade ou outra técnica de raytracing mais avançada, após o processamento de todos os polígonos da cena, o buffer de intensidade/cor resultante representará exactamente os valores dos pixels da Viewport e pode ser visualizado sem mais nenhuma transformação.

O algoritmo Z-Buffer trabalha em Coordenadas de Vídeo ou Viewport (espaço de dispositivo) e pode ser implementado facilmente como uma extensão/modificação de um algoritmo de conversão por varredura. O Z-Buffer ou Buffer de Profundidade, estrutura de dados da qual o método tira seu nome, é uma matriz $[n \times n]$ na qual o elemento (i,j) corresponde ao pixel (i,j) . Esta matriz contém o valor de z em Coordenadas do Mundo (espaço de imagem) do objeto correntemente visível naquele pixel. Além disso, existe outra matriz $[n \times n]$ cujos elementos correspondem à cor ou intensidade luminosa que é para ser atribuída àquele pixel.

O próximo passo é mapear, pixel a pixel, a profundidade de cada objecto da cena que se encontra atrás daquele pixel. Toma-se cada um dos polígonos, executa-se a conversão por varredura e, durante esta conversão, executa-se os passos seguintes:

1 - Verificação de intersecção. Para cada píxel, determina-se os polígonos da cena cuja conversão de varredura resultou num elemento com as coordenadas x,y deste pixel.

2 - Determinação da ordem de ocultação. Para todos os polígonos da cena, determina-se para cada pixel (i,j) que intercepta o polígono, se o valor z em Coordenadas do Mundo (eixo z do desenho acima) deste polígono na posição do pixel é maior do que o valor z armazenado no Z-Buffer. Se for maior, calcula-se a cor e outros valores associados a este pixel e insere-se na entrada (i,j) da matriz de cor e insere-se o valor z do polígono naquele ponto na posição (i,j) do z-Buffer. Senão, não se faz nada.

Pseudocódigo simples:

```
Algoritmo Z-Buffer
Dados:
Lista of polígonos {P1, P2, ..., Pn}
Matriz z-buffer[x,y] inicializado com  $\infty$ 
Matriz Intensidade[x,y]
Início
para cada polígono P na lista de polígonos faça {
    para cada pixel (x,y) que intercepta P faça {
        calcule profundidade-z de P na posição (x,y)
        se prof-z < z-buffer[x,y] então {
            Intensidade[x,y] = intensidade de P em (x,y)
            z-buffer[x,y] = prof-z
        }
    }
}
Desenhe Intensidade
fim
```

Uma vez que todo polígono pode ser dividido num conjunto de trapézios (se se aceitar o triângulo como um trapézio degenerado de topo zero), basta-nos considerar o algoritmo Z-buffer sobre trapézios. Para tal pode-se directamente modificar o algoritmo de renderização, ou então um dos algoritmos de rasterização mais avançados. Pode-se utilizar directamente os procedimentos de rasterização e de actualização de vídeo implementados pelo algoritmo de conversão que for ser utilizado.