



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Computação Gráfica

AP2 - 1º semestre de 2008.

Nome –

Assinatura –

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
 2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
 3. Você pode usar lápis para responder as questões.
 4. Ao final da prova devolva as folhas de questões e as de respostas.
 5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
-

- 1) O Algoritmo de Ray-tracing permite geração de imagens realistas. Entretanto, é uma solução inviável para computação gráfica tempo real. Justifique.

O Ray tracing é um algoritmo onde para cada pixel da tela é traçado um raio, calcula-se a interseção do mesmo com os objetos geométricos da cena. Quando há interseção, ocorre a chamada recursiva para um raio de reflexo e outro de transparência, simulando com precisão os fenômenos óticos da luz da cena. Assim, o algoritmo é de $O(n!)$, onde n é o número de elementos da cena. Isto torna totalmente inviável o processamento em tempo real.

- 2) Porque o componente especular, no modelo de iluminação *per vertex*, não é preciso?

Na iluminação por vértice, calcula-se a iluminação apenas dos vértices. A iluminação do interior dos polígonos é então calculada por interpolação das cores obtidas. O coeficiente especular é restrito a apenas pequenas regiões da superfície. Ao fazer a interpolação, o componente especular será completamente ignorado. Caso o vértice possua especular, esta iluminação irá se espalhar na superfície de uma forma não real, pois a interpolação é linear no interior do polígono.

- 3) Descreva a em detalhes o componente de Iluminação Especular no modelo *Phong*.

A componente especular consiste em: $K \cdot (O \cdot R)^n$

K é o coeficiente ou cor da superfície em questão

O é o vetor normalizado que aponta do observador para a superfície em questão.

R é o vetor reflexo normalizado, que consiste no reflexo do vetor luz em relação a normal da superfície

N é o coeficiente de especularidade. Quanto maior este coeficiente, o produto escalar de O.R se aproximará com maior velocidade a zero, exceto no caso de valer 1. (0 graus entre O e R) O efeito deste coeficiente é de que quanto maior for o n, mais concentrado está o especular num único ponto da superfície.

4) Descreva o algoritmo Sutherland-Hodgman para recorte de polígonos.

O algoritmo de *Sutherland* e *Hodgman* é semelhante ao algoritmo de *Cohen-Sutherland*. O polígono é recortado sucessivamente contra todos os lados da figura de recorte, ou seja, a janela de recorte retangular. O algoritmo trabalha sobre uma lista circular de vértices, sendo que os vértices, e também as arestas que os conectam, são processados em seqüência e classificados contra o lado corrente do polígono de recorte.

Criar um lista circular **le** contendo os vértices do polígono.

PARA (cada lado **l** da janela)

PARA (cada vértice **v** de **le**)

SE (**v** está dentro da janela de recorte) ENTÃO

Copiar **v** para a lista de saída **ls**

FIM_SE

SE (a aresta formada por **v** e o sucessor **v'** de **v** intersecta **l**) ENTÃO

Calcular ponto de interseção **v''**.

Copiar **v''** para a lista de saída **ls**.

FIM_SE

FIM_PARA

le ← **ls**

FIM_PARA

- 5) O que são o *Front-Buffer* e o *Back-Buffer*? Como eles funcionam durante o processo de renderização?

Front Buffer é a memória de vídeo principal. Tudo o que se escreve nela, aparece na tela. O Back buffer é uma memória secundária. Caso a renderização fosse feita diretamente no front buffer, seria possível ver a imagem sendo formada durante a aplicação. Isto na prática causa um ruído indesejado. Na prática a renderização é totalmente feita no second buffer e apenas copiado para o front buffer – e portanto mostrada na tela – quando a imagem já estiver pronta.