

Prova 2 – Computação Gráfica

Em cada resposta, procure utilizar ao máximo equações, figuras e comentar assuntos que sejam diretamente relacionados. Para cálculo de ângulos, utilize produto escalar. Dê respostas objetivas.

2. 1. Explique como o algoritmo ray tracing consegue gerar sombras e o mesmo não ocorre com o OpenGL, exceto com a adição de recursos extras.
- 10 2. Explique como é calculada a iluminação difusa em uma solução que utiliza o algoritmo z-buffer (Ex: OpenGL, Canvas2D). Elabore um algoritmo abstrato para pintar um modelo 3D.
3. ~~Desenhe, comente e compare as blending functions de Bézier de grau 3 com a Spline de grau 3 para geração de superfícies. Considere a geração de um único patch.~~
- 10 4. Explique o que é super amostragem em um algoritmo de ray casting, para que serve, os tipos existentes, os problemas que ocorrem se não for utilizado, e a computação final da cor do pixel.

Resposta da 4.

→ O que é super amostragem?

É quando para cada pixel do sensor da câmera são projetados vários raios em ao invés de somente um.

→ Para que serve?

Serve para amostrar mais precisamente a cor final final no sensor da câmera, quando os raios colidem em superfícies com muita variação de contribuição, por exemplo, a longas distâncias.

→ Tipos existentes

- Não sei. Chutei:
1. vários raios projetados com ângulos equiespaçados
 2. vários raios projetados com ângulos aleatórios
 3. vários raios projetados em tempos diferentes
(gera blur)

→ Problemas se não utilizados

Artefatos visuais, Por exemplo, um tabuleiro de xadrez visto de longe pode ficar todo distorcido, pois ao se usar um único raio, raios adjacentes podem estonteantemente colidir com vários quadrados brancos ou pretos, ao invés de sempre intercalado branco com preto

→ Computação final da cor: Não é simples, se fizer média, fica todo borrado, pode-se

Fazer média de setores e aí combiná-los, ou fazer chutinhos só lembrar que não é trivial

Resposta da 1.

Raytracing é capaz de gerar sombras, pois para cada colisão de um raio projetado da câmera, é necessário calcular a contribuição local, que por sua vez, conforme a Figura 3, vai projetar inúmeros outros raios em outras direções (sujeitos a reflexão e refração, conforme espelho da Figura 3),

e estes, caso colidam com um objeto luminoso, ficarão claros, caso contrário formam uma sombra.

Na Figura 1, a contribuição local da sombra de d é porque dali nenhum raio atingiu nenhuma fonte luminosa, a sombra f atingiu uma e em e , apesar de um raio gerar sombras, o outro colidiu com o feixe, anulando a sombra.

A Figura 2 demonstra que no cálculo da contribuição local, os raios podem passar por superfícies refletoras antes de atingir (ou não) uma fonte luminosa.

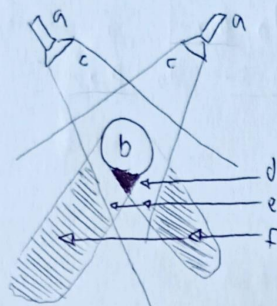
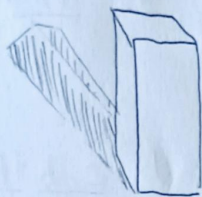


Figura 1.

- a. fonte luminosa
- b. cilindro visto de cima
- c. feixes de luz
- d. sombra total
- e. projeção da sombra de uma luz ofuscada pela outra fonte de luz
- f. sombra de um feixe

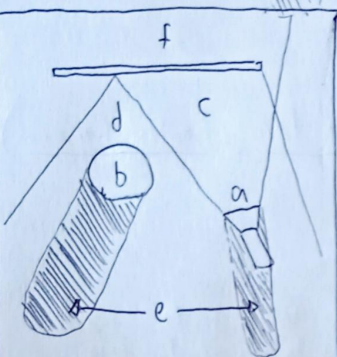


Figura 2

- a. fonte luminosa
- b. cilindro visto de cima
- c. feixe de luz
- d. feixe de luz refletido
- e. sombra
- f. espelho

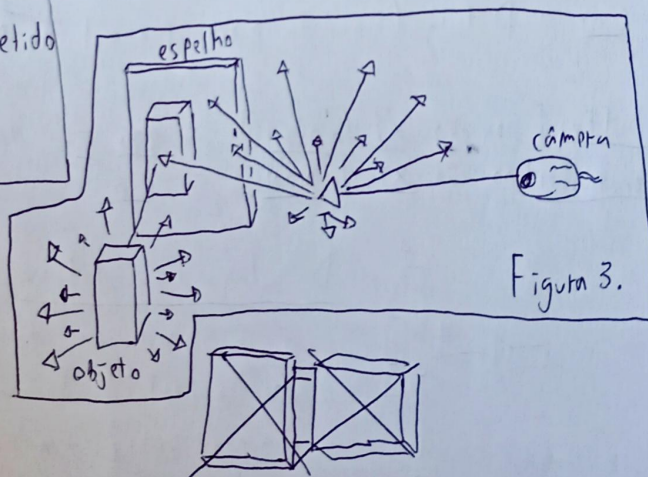


Figura 3.

Resposta da 1.

Cont.

OpenGL não projeta sombras, pois meramente toma os vetores ~~das~~ normais das superfícies e as fontes luminosas, e calcula a iluminação local (Gaudier, Phong, Flat), toda a lógica do raytracing está ausente, ~~de modo que~~ as sombras não são levadas em consideração, simplesmente.

Resposta da 2.

Scanline (Objetos, Fontes Luminosas);

Para cada face de todas as faces de todos os objetos 3D da cena;

Para cada pixel da face;

Se componente z do pixel é menor que o z do z-buffer correspondente na posição do pixel;

Pula;

Senão;

Caso iluminação Flat;

$$\underline{Cor} = \underline{Difusa}(\underline{Cor\ da\ face}, \underline{Normal\ da\ face}, \underline{Fontes\ luminosas})$$

Caso iluminação Gaudier;

~~$$\underline{Cor} = \underline{Difusa}(\underline{Cor\ em})$$~~

$$\underline{Cor} = \underline{Interpolação-bilinear}(\underline{Difusa}(\underline{Cor\ do\ Vértice\ da\ face}, \underline{Normal\ do\ Vértice\ da\ face}, \underline{Fontes\ luminosas}), \dots)$$

Caso iluminação Phong;

$$\underline{Cor} = \underline{Difusa}(\underline{Cor\ do\ pixel}, \underline{Normal\ do\ pixel}, \underline{Fontes\ luminosas})$$

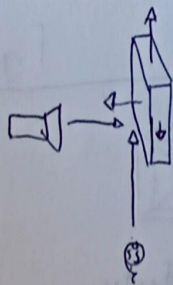
$$\underline{z-buffer[posição\ XY\ do\ pixel]} = \underline{z\ do\ pixel}$$
$$\underline{framebuffer[posição\ XY\ do\ pixel]} = \underline{Cor}$$

End

End

End

Return framebuffer



Da iluminação difusa;

é a iluminação de objetos opacos, com superfície fosca, toma o ângulo entre a normal e a fonte luminosa, próximo de 0° a diferença entre ambas, brilho alto, quanto maior o ângulo, mais escuro, preto para diferenças perto de 180°