


P1 CG 2014.1

	Computação Gráfica 1 Prof. Rodrigo de Toledo Prof. Luis Peñaranda	Data: 16/04/2014 P1 2014.1
---	--	-------------------------------

1) (3 pontos) (visualização volumétrica)

A figura abaixo representa uma fatia de um dado volumétrico, para se calcular a cor final de um determinado pixel, um raio é atirado contra o volume o atravessando como no desenho abaixo.

A partir do dado volumétrico na figura abaixo e sua função de transferência, responda aos seguintes itens.

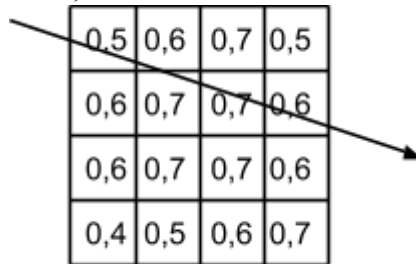
Considere que a cor opaca da iso-superfície foi definida pelo usuário como [1,0,0].

a) reproduza a matriz na sua folha de respostas (sem os valores). Desenhe a iso-superfície de valor 0.65. Dicas: (1) Lembre-se que o valor equivale ao **centro** de cada célula; (2) Lembre-se do algoritmo de Marching Cubes, pode ajudar.

Considere que o cosseno entre a direção da luz e a normal no ponto interceptado da iso-superfície é 0.5.

Qual a cor final do pixel nas seguintes situações:

- b) Iso-superfície com iluminação apenas ambiente (coeficiente ambiente constante = 0.5);
- c) Iso-superfície com iluminação ambiente + difusa (coeficiente ambiente constante = 0.5 e ambiente = 0.5);
- d) Iso-superfície com iluminação ambiente + difusa com sombra (a luz é direcional [0,-1], ou seja, vem de baixo).
- e) Calcule o valor do pixel acumulado pelo raio, usando a fórmula vista na aula.



Função de transferência de opacidade:
$$T(x) = \begin{cases} 0 & \text{se } V(x) < 0.5 \\ 2 * (V(x) - 0.5) & \text{se } V(x) \geq 0.5 \end{cases}$$

2) (2 pontos) Faça um programa (pode usar sintaxe GLSL, apesar de ser algo que executará em CPU) que a partir de uma textura representando um mapa de altura, gere uma textura representando o mapa de normais (usando a informação de vizinhança). Você pode usar comandos comuns na linguagem GLSL como dot, cross, normalize, e vec3.

```
int height = 150;
int length = 300;
vec3 heightMap[height][length];
vec3 normalMap[height][length];
void main (void) {...}
```

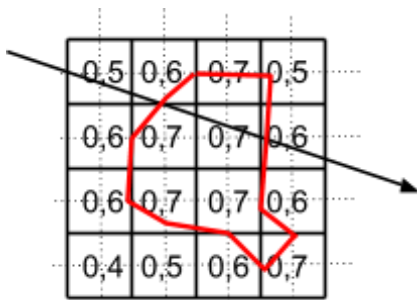
3) (1.5 pontos) (P2 2011.1) Cite três problemas do algoritmo do pintor. Qual deles é o pior e por que? (A resposta deve ter no máximo 50 palavras.)

4) (1.5 pontos) (P2 2012.2) Suponha uma cena que contenha uma malha de dezenas de milhares de triângulos em uma posição fora do campo de visão da câmera. O algoritmo de Ray Tracing implementado é lento para renderizar esta cena, mesmo que a imagem final não contenha nenhum objeto. Cite uma técnica de aceleração do algoritmo que otimizaria a renderização da cena descrita, explicando sua razão.

5) (2 pontos) Por que a geometria afim não é capaz de modelar as projeções? Descreva uma solução para este problema.

GABARITO:

1) a)



3) Os três problemas são:

1. eficiência: desenhar objetos invisíveis innecessariamente;
2. consistência: a superposição dos objetos não é consistente;
3. ordenação: como determinar a ordem no eixo Z dos objetos?

Publicado por [Google Drive](#) – [Denunciar abuso](#) – 5Atualizado automaticamente a cada minutos
