

Trabalho II

1 Especificação do Problema

Em computação gráfica, superfícies são normalmente representadas por meio de um conjunto de polígonos, denominado malhas poligonais. Devido a sua simplicidade, triângulos são primitivas gráficas comuns utilizadas em malhas poligonais.

O objetivo deste trabalho é manipular e visualizar modelos representados por malhas triangulares. O programa deve utilizar as bibliotecas OpenGL e GLUT, as quais deverão ser incorporadas em um programa escrito em linguagem de programação C, C++ ou Java.

Seu programa deve inicialmente carregar um modelo no formato (simplificado) Wavefront OBJ, o qual contém informações sobre os vértices e as faces das malhas triangulares. Uma estrutura de dados deve ser criada para armazenar a malha, mantendo-se a sua geometria e topologia.

As faces da malha triangular devem ser desenhadas nos seguintes estilos:

- estrutura de arame (*wireframe*): apenas as fronteiras (bordas ou arestas) dos polígonos devem ser desenhadas;
- sombreado plano (*flat shading*): cada triângulo deve possuir uma única cor, tal que a superfície do objeto terá uma aparência facetada;
- estrutura de arame + polígonos: as arestas de cada polígono devem ser sobrepostas às faces da malha;
- sombreado suave (*smooth shading*): a superfície possui cor e suavização. Cada vértice pode ter uma cor associada a ele. As técnicas comuns para suavização são Gouraud e Phong;
- silhueta: as silhuetas de um objeto são as arestas onde a superfície é tangente à direção de visualização. Isto corresponde às áreas comuns entre as faces ocultas (*back facing*) e as faces visíveis (*front facing*) dos polígonos. Para obter as silhuetas, uma maneira é desenhar os polígonos com uma determinada cor e então desenhar as faces ocultas, selecionadas com a primitiva *glCullFace* e a estrutura de arame com bordas espessas. A espessura da linha pode ser alterada com a primitiva *glLineWidth*. Então, as partes da estrutura de arame serão mostradas apenas onde há uma silhueta.
- silhueta + sombreado: as silhuetas do objeto devem ser sobrepostas no modelo sombreado.

Esses estilos de desenho são ilustrados na figura 1. Cada um dos estilos deve ser individualmente especificado pelo usuário por meio da interface gráfica. O programa deve também permitir movimentação do ponto de observação (câmera) ou ser capaz de aplicar transformação de rotação ao modelo.

2 Formato OBJ

O formato OBJ foi desenvolvido pela Wavefront para especificar objetos representados por malhas poligonais. O formato apresenta inicialmente uma lista de vértices seguida de uma lista de faces.

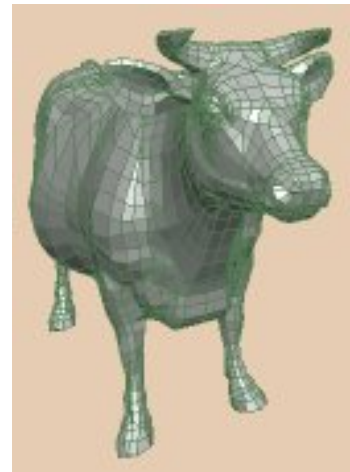
O caracter *v* especifica um vértice, enquanto o caracter *f* especifica uma face. Um exemplo de representação de um cubo é mostrado a seguir. Como todas as faces são triangulares, elas são formadas pelas coordenadas de três vértices. A face 1, por exemplo, é formada pelos vértices 1, 2 e 3 da lista de vértices.



(a) estrutura de arame



(b) sombreamento plano



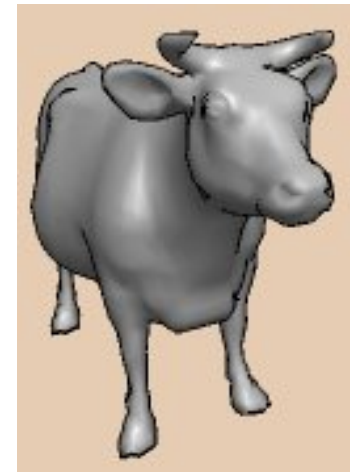
(c) estrutura de arame + polígonos



(d) sombreamento suave



(e) silhueta



(f) silhueta + sombreamento

Figura 1: Modelo desenhado em diferentes estilos.

```

v -0.5 0.5 0.5
v -0.5 -0.5 0.5
v 0.5 -0.5 0.5
v 0.5 0.5 0.5
v -0.5 0.5 -0.5
v -0.5 -0.5 -0.5
v 0.5 -0.5 -0.5
v 0.5 0.5 -0.5
f 1 2 3
f 8 7 6
f 4 3 7
f 5 1 4
f 5 6 2
f 2 6 7
f 1 3 4
f 8 6 5
f 4 7 8
f 5 4 8
f 5 2 1
f 2 7 3

```

3 Exemplos de Modelos

Alguns modelos no formato OBJ estão disponíveis em:

http://www.ic.unicamp.br/~helio/modelos_obj/

4 Documentação das Bibliotecas

Alguns manuais das bibliotecas OpenGL e GLUT podem ser encontrados nos endereços:

<http://www.opengl.org>

<http://www.glprogramming.com/red>

<http://www.sgi.com/products/software/opengl>

<http://www.lighthouse3d.com/opengl/glut/>

5 Exemplos de Códigos OpenGL

Alguns códigos escritos em OpenGL estão disponíveis em:

http://www.ic.unicamp.br/~helio/codigos_opengl/

6 Especificação da Entrega

- A entrega do trabalho deve conter os seguintes itens:

- código fonte: o arquivo final deve estar no formato *zip* ou no formato *tar* compactado com *gzip*, contendo todos os programas necessários para sua execução. Favor incluir um Makefile para facilitar a compilação dos programas.
- relatório impresso: deve conter uma descrição dos algoritmos e estruturas de dados, considerações adotadas na solução do problema, testes executados, eventuais limitações ou situações especiais não tratadas pelo programa.

- Data de entrega: 30 de setembro de 2009

7 Observações Gerais

- Os programas serão compilados em ambiente Linux. Os formatos de entrada e saída dos dados devem ser rigorosamente respeitados pelo programa, conforme definidos anteriormente. Trabalhos entregues com atraso terão 10% da nota descontada por dia de atraso. Não serão aceitos trabalhos após 5 dias da data de entrega.
- Os seguintes aspectos serão considerados na avaliação: funcionamento da implementação, clareza do código, qualidade do relatório técnico.