

COMPUTAÇÃO GRÁFICA - TRABALHO 1

Willian Henrique Siman de Lima

Maio de 2018

Detalhes da Implementação

Partindo da imagem de entrada, cada coordenada de cada pixel $(x, y, f(x, y))$ foi normalizada e ajustada para que o objeto estivesse centralizado em $(0, 0, 0)$. Essas coordenadas foram inseridas sequencialmente num vetor. Esse vetor contém todos os vértices da nuvem de pontos.

O modelo da malha de triângulos foi computado a partir do índice de cada vértice no vetor da nuvem de pontos. Os vértices foram inseridos em ordem tal que o vértice 0 é o pixel no topo esquerdo da imagem original, o vértice à direita do vértice 0 é o vértice 1, e o vértice abaixo do vértice 0 é o vértice com índice igual à largura da imagem de entrada.

Para cada vértice (respeitando os limites da imagem), um triângulo era formado entre ele mesmo, o vértice à sua direita e o vértice de baixo e outro triângulo era formado entre o vértice de baixo, o vértice embaixo e à direita e o vértice à direita.

Cada um desses triângulos nada mais é do que 3 elementos inseridos em sequência num vetor. Esse vetor foi inserido num Element Buffer Object, que é efetivamente usado para desenhar o objeto.

Exemplos de uso das transformações

A translação pode ser usada para se aproximar e afastar de pontos do modelo. Rotações podem ser usadas para ver o modelo sob outro ângulo. A transformação de escala pode ser usada como uma espécie de zoom, sem aproximar o objeto do plano de projeção necessariamente, mas tornando-o maior de toda forma.

Escolhas feitas para gerar a visualização

As cores do fundo e dos elementos desenhados foram escolhidas de forma que nenhum elemento fosse difícil de enxergar. Por simplicidade, as cores de cada elemento são função direta de sua posição no espaço ($x \rightarrow R, y \rightarrow G, z \rightarrow B$). Como pontos próximos à origem ficavam escuros, a cor de fundo foi ajustada para um tom de cinza escuro.

Todas as transformações aumentam ou diminuem linearmente com cada comando, diminuindo a sensibilidade ao tempo de resposta do usuário em relação as suas ações. Os fatores foram escolhidos de forma que sejam necessários 20 comandos de translação para ir de uma ponta a outra da imagem num mesmo eixo, 20 comandos de rotação em um mesmo eixo para obter rotação de 180° e o fator de escala foi escolhido empiricamente, com foco em transformações de até 2x ou 3x (um fator de escala por comando maior seria mais adequado caso seja desejado trabalhar mais frequentemente com grandes variações de escala).

Todas as transformações são aplicadas de forma independente por questão de simplicidade. Rodar um objeto ao redor do eixo X não levará em consideração sua rotação em relação à câmera. Similarmente para a escala, transformações de escala aumentarão e diminuirão o objeto em relação as suas próprias coordenadas (e não em relação à câmera). A translação é a única transformação tomando a câmera como referência.

A projeção usada foi uma projeção em perspectiva, com proporção de tela 16/9 e 90° de campo de visão vertical. A projeção em perspectiva provê um sentimento de estar navegando ao redor do objeto de fato, em contrapartida a projeções ortogonais, que removem a noção de distância. O campo de visão foi escolhido com base em valores padrão de projeção.

Falhas e dificuldades

O programa não apresenta nenhuma falha notável: ele cumpre os requisitos pedidos adequadamente. Uma pequena falha que pode-se apontar é o fato da razão de proporção da imagem original se perder no processo de normalização. Todos os pontos são normalizados para um cubo unitário. Essa foi uma decisão feita devido a dificuldades em projetar os pontos sem normalizá-los. Inicialmente uma matriz de projeção foi usada para tentar colocar todos os pontos do objeto dentro do campo de visão, mas sem muito sucesso.

Computar a malha de triângulos também apresentou uma leve complexidade. Inicialmente, muitos mais triângulos estavam sendo computados, formando uma grande quantidade de artefatos visuais. Esta falha foi causada por percorrer os índices da imagem de forma inadequada.

Outras melhorias que poderiam ser feitas (não se caracterizando como uma falha, mas simplesmente uma nova função/característica a ser adicionada) seriam a alteração das cores dos elementos desenhados, permitir a troca entre projeção ortogonal e de perspectiva, um *reset* de todos os parâmetros alterados e permitir a rotação apenas da câmera, independente do objeto.