

RECONSTRUÇÃO GRÁFICA TRIDIMENSIONAL DE EDIFICAÇÕES URBANAS A PARTIR DE IMAGENS AÉREAS

Daniel Moisés Gonzalez Clua¹, Dr. Valdemir Carrara²

¹Universidade do Vale do Paraíba, Av. Shishima Hifumi, 2911 – Urbanova
São José dos Campos, SP, 12244-000

²Instituto Nacional de Pesquisas Espaciais – INPE / Divisão de Mecânica Espacial e Controle – DMC
Av dos Astronautas, 1.758 – Jardim da Granja
São José dos Campos, SP, 12227-010
dmgc29785@yahoo.com.br⁽¹⁾, val@dem.inpe.br⁽²⁾

Resumo - Este trabalho, iniciado em abril de 2005, tem como objetivo o desenvolvimento de algoritmos para a obtenção das dimensões de construções de edifícios a partir do processamento de imagens aéreas de alta resolução. Estas dimensões serão posteriormente utilizadas para compor objetos gráficos tridimensionais utilizando a própria textura obtida da imagem. No presente trabalho, os vértices das construções deverão ser fornecidos por um dispositivo apontador. O trabalho compreende também a eliminação da paralaxe, a correção da iluminação em função do ângulo de elevação solar, e a eliminação de sombra na textura. A metodologia a ser utilizada no desenvolvimento da presente proposta envolve a aplicação de geometria analítica e vetorial no desenvolvimento de algoritmos para compor as dimensões das construções tridimensionais. O algoritmo será desenvolvido em linguagem C ou C++, com visualização realizada por meio de OpenGL.

Palavras-chave: Computação Gráfica, OpenGL, Reconstrução de Edifícios, Ambientes Virtuais.

Área do Conhecimento: Ciências Exatas, Ciência da Computação, Computação Gráfica

Introdução

A construção de ambientes gráficos tridimensionais (ambiente virtual) no computador tem encontrado diversas aplicações recentemente. Embora ambientes virtuais sejam freqüentes em jogos, cenários tridimensionais que representam ambientes reais são cada vez mais empregados no turismo, no urbanismo, no tratamento de distúrbios nervosos (fobias) e simuladores de voo, veículos e máquinas. Também é usual, em tais ambientes, a reconstrução virtual de partes de cidades ou mesmo cidades inteiras. Uma vez que esta reconstrução pode ser bastante complexa e difícil, em virtude dos vários cenários existentes, buscam-se ferramentas automáticas ou semi-automáticas para tal reconstrução (VTP 2005, Carmenta 2005). Imagens aéreas e imagens obtidas por satélites com alta resolução são utilizadas usualmente, o que permite reconstruir grandes áreas com poucas imagens. O processo de reconstrução virtual pode ser dividido em duas partes: na primeira buscam-se formas automáticas ou semi-automáticas de extração de características (ou informações) das imagens para detectar as construções, e na segunda efetua-se a reconstrução propriamente dita. Pretende-se neste trabalho investir na reconstrução de ambientes urbanos (edifícios e casas) a partir de informações previamente fornecidas a um programa

computacional, o que caracteriza um método semi-automático.

Portanto, o objetivo deste trabalho é elaborar um programa que permite reconstruir espacialmente e visualizar construções e edifícios a partir de imagens aéreas. A linguagem usada é C++, com recursos de OpenGL para a visualização.

O OpenGL é definido como “uma interface em software para hardware gráfico”. Em essência, é uma biblioteca de gráficos tridimensionais e modelagem, portátil e rápida, o que permite gerar imagens interativas em tempo-real. O OpenGL não é uma linguagem, como C ou C++, mas sim uma biblioteca de funções pré-programadas. Na realidade não existe um “programa em OpenGL” mas sim um programa que o desenvolvedor escreve usando o OpenGL como um de seus APIs. As informações coletadas para a configuração adequada do OpenGL e para a preparação do programa vieram do livro OpenGL Super Bible (Wright, 2000) e de programas tutores encontrados na Web (Neon-Helium Productions, 2005).

Materiais e Métodos

A interface do programa consiste em uma janela principal com duas visualizações: uma em perspectiva e uma em paralelo. A visão em

perspectiva cria uma distorção nos objetos, conforme eles se afastam, dando uma noção de profundidade. Este tipo de visualização dá um maior realismo à cena e é utilizado para exibir os edifícios construídos, permitindo uma movimentação livre, tanto em translação como em rotação. Já a vista superior em projeção paralela é utilizada para a entrada de pontos pelo usuário, já que não existe distorção e, portanto, não há risco de ocorrerem erros em consequência disso.

Além desta janela principal, o programa apresenta um menu e botões que permitem mudar o tipo do cursor (para rotação, translação, etc.), construir um edifício, salvar ou carregar a cena e sair do programa.

As principais funções implementadas para efetuar a reconstrução virtual são:

1. Fornecimento dos vértices referentes ao topo do edifício;
2. Fornecimento do vetor para translação;
3. Fornecimento do vetor da sombra, para cálculo de altura;
4. Extrair a textura a ser aplicada no topo do edifício da própria imagem;
5. Usar uma imagem adicional para 'simular' a textura das laterais do edifício;
6. Minimizar a sombra do edifício na textura do solo;
7. Exibir o resultado na forma tridimensional.

Fornecimento dos vértices e vetores

Na implementação atual, o usuário deve fornecer um número qualquer de pontos para o topo do edifício (como por exemplo, os pontos P1 a P12, da figura 1), dois pontos que representem o vetor da altura do edifício (h) e mais dois pontos que representam o vetor da sombra (s). Uma vez que a leitura das coordenadas destes pontos, efetuada pelo mouse, estão em unidades de pixel da janela, é então necessário convertê-las para unidades de pixel da imagem por meio de uma transformação linear:

$$Tx_i = a \cdot x_i + dx$$

$$Ty_i = a \cdot y_i + dy$$

onde a é a ampliação (fator de escala) utilizada na visualização. Estes valores são então atribuídos a dois vetores, um para as coordenadas no eixo x e outra para as do eixo y . Ambos possuem os valores de todos os pontos que representam o topo do edifício e também os dos vetores altura e da sombra. O vetor da altura serve para calcular o deslocamento que o programa deve aplicar ao objeto para que ele se origine da base do edifício na imagem e não do topo. Já o vetor da sombra é necessário para calcular a altura do edifício, através de um cálculo com o ângulo de elevação

do sol (figura 2). Passados estes pontos, o programa constrói o objeto que representa o edifício, usando os valores armazenados das coordenadas..

Para possibilitar a reconstrução de vários edifícios, foi criada uma classe para representá-los. Deste modo, podem ser instanciados diversos objetos desta classe, sendo que cada um contém todas as suas informações necessárias, como a sua posição no plano, os vértices entrados pelo usuário, a altura, etc. Além de facilitar a construção dos edifícios na cena, a classe irá permitir que, mais adiante, o usuário possa selecionar um edifício já construído e modificá-lo.



Fig. 1 – Pontos recebidos para a construção do edifício.

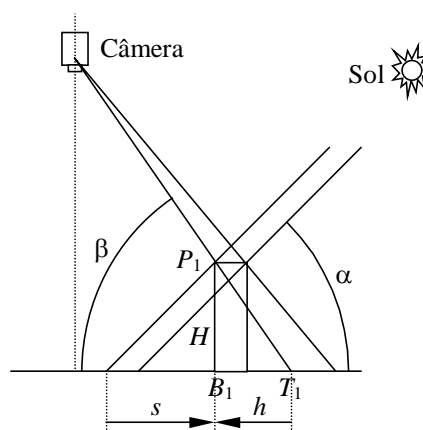


Fig. 2 – Vista lateral da geometria do problema.

Triangularização

Como mencionado acima, o edifício consistirá de um número qualquer de vértices, resultando em um polígono que pode ter qualquer formato. Porém, o OpenGL apenas trabalha com triângulos. Ou seja, todos os objetos tridimensionais construídos na tela, nada mais são do que vários triângulos juntos. Deve-se, então,

“triangularizar” o polígono resultante da entrada de vértices: dividi-lo em triângulos.

Para tal objetivo, foram desenvolvidos diversos algoritmos para que o triângulo satisfaça algumas condições necessárias, como estar no sentido anti-horário (figura 3) e remover pontos sobrepostos ou “supérfluos” (inseridos numa aresta, fazendo com que haja 3 ou mais pontos em seqüência alinhados a uma mesma reta). Tendo essas condições, o programa segue para o algoritmo principal, que divide o polígono em triângulos (figura 4), usando um critério recursivo de remoção dos triângulos internos.

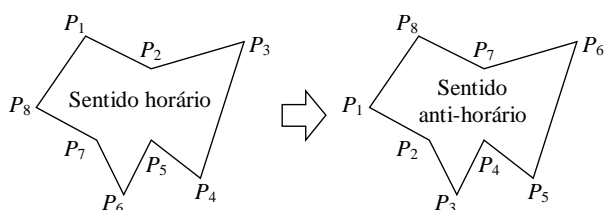


Fig. 3 – Inversão do sentido dos pontos que definem o polígono.

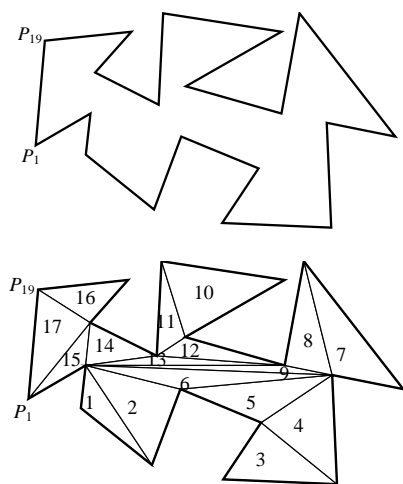


Fig. 4 – Subdivisão do polígono em triângulos.

Aplicação de texturas

A aplicação de texturas permite melhorar o realismo do ambiente virtual pela aplicação de uma imagem à superfície de objetos. Para isto, é necessário utilizar várias funções do OpenGL para carregar uma imagem, transformá-la em textura, armazená-la em uma variável e, finalmente, aplicá-la a algum objeto.

No edifício reconstruído, há dois tipos diferentes de aplicação de texturas: no topo e nas laterais. Para o topo, pode-se utilizar a imagem original do solo, já que as partes superiores dos edifícios estão quase sempre bem visíveis (figura 5). Já para as laterais isto geralmente não acontece, então usam-se imagens adicionais que

“simulam” possíveis padrões de paredes e janelas de edifícios (figura 6).



Fig. 5 – Aplicação da imagem original no topo do edifício

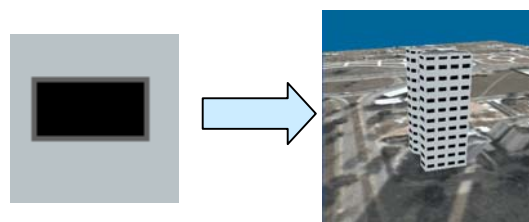


Fig. 6 – Aplicação da imagem adicional nas laterais do edifício

Minimização da sombra na textura do solo

Uma vez que a reconstrução dos edifícios é realizada graficamente, deseja-se que a iluminação na cena reconstruída possa ser alterada sob ação externa. Em outras palavras, deseja-se que as sombras dos objetos na cena acompanhem o movimento da posição da fonte de luz artificial quando esta for modificada, de modo que a projeção tenha um aspecto real. Isto cria um problema, pois em geral as imagens aéreas que servem como textura do solo apresentam sombras naturais dos edifícios. Para que não houvesse uma “sobreposição” de sombras (uma natural e outra artificial), o que causaria uma indefinição na direção real de iluminação, foi necessário encontrar uma maneira de se “eliminar” a sombra natural. Admitindo-se que a área que a sombra ocupa no solo (polígono de sombra) seja conhecida, pode-se reaplicar a própria textura do solo a este local com uma claridade maior que a do solo. Isto é feito elevando-se os valores de cores vermelha, verde e azul de cada pixel. As quantidades elevadas em cada canal de cor são definidas pelo usuário e devem ser ajustadas empiricamente.

Aplicando-se a quantidade certa de brilho, obtêm-se o desaparecimento praticamente completo da sombra na imagem original, possibilitando a projeção de sombras dinâmicas na cena (ainda não implementado atualmente). Na presente fase do projeto, o usuário deve indicar os vértices do polígono da sombra. Essa entrada

funciona de maneira semelhante à entrada de vértices do edifício, e o polígono passa pelo processo de triangularização de forma idêntica àquela realizada no topo. A figura 6 mostra um exemplo no qual a sombra do solo foi minimizada. Percebe-se, neste exemplo, que a completa eliminação da sombra é algo difícil, senão impossível de ser realizada. Contudo a redução do seu efeito na textura do solo irá permitir que a direção de iluminação artificial seja evidenciada sem ambigüidade.

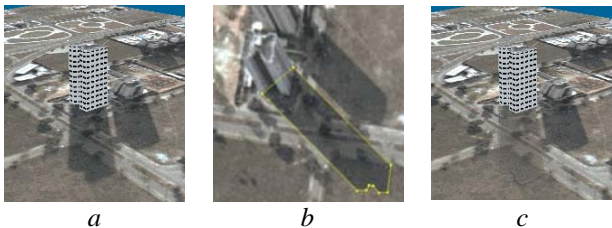


Fig. 6 - Exemplo de edifício com a sombra original (a), seleção da área da sombra (b) e eliminação da sombra (c).

Resultados e discussão

Tendo as posições de todos os vértices referentes à base do edifício, sabendo sua altura e com todas as texturas prontas, o edifício é reconstruído tridimensionalmente na visão em perspectiva do programa, a qual é possível movimentar livremente.

Na implementação atual, podem ser construídos vários edifícios, os quais são construídos a partir de polígonos com qualquer formato, apresentam texturas tanto na parte superior como nas laterais e têm suas sombras originais removidas. As cenas criadas podem ser salvas e carregadas. Ainda não há um processo de iluminação e os edifícios não podem ser alterados depois de construídos.

A figura 7 mostra exemplos de resultados obtidos atualmente:

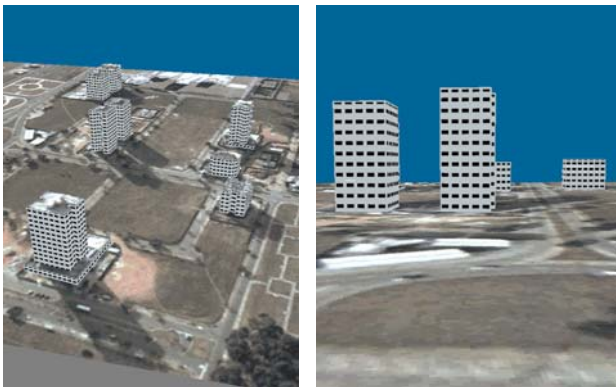


Fig. 7 – Resultados atuais.

Conclusão

Nesta fase parcial de desenvolvimento considera-se que os objetivos, que consistiam de familiarização com as funções do OpenGL, construção dos edifícios através das entradas do usuário e algumas modificações na textura do solo foram atingidos. Os próximos aprimoramentos do trabalho podem ser resumidos em:

- Iluminação dinâmica da cena
- Possibilidade de modificar um edifício já construído
- Remoção da imagem do edifício na textura

Referências

- VTP - *The Virtual Terrain Project*, 2005. Disponível em <http://www.vterrain.org/>. Acesso em 31 jul. 2006.
- Carmenta – *SpatialAce*, 2005. Disponível em <http://www.spatialace.com/>. Acesso em 31 jul. 2006.
- Neon-Helium Productions, *NeHe*, 2005 Disponível em <http://nehe.gamedev.net/>. Acesso em 31 jul. 2006.
- Wright, S. R.; Sweet, M. *OpenGL Super Bible. Second Edition*. Waite Group Press, 2000.