

FAPESP – FUNDAÇÃO DE AMPARO À PESQUISA DO ESTADO DE SÃO PAULO

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA (IME)

RELATÓRIO FINAL

GeoGrid

VISUALIZAÇÃO TRIDIMENSIONAL DE TERRENO

Integrantes:

Felipe Augusto A. Dias

Franklin de Oliveira

Giseli de A. Ramos

João Pedro Kerr Catunda

Vivian L. Cury

Orientação do Professor Doutor Roberto Marcondes César Júnior.

ÍNDICE

1. INTRODUÇÃO	3
2. OBJETIVOS.....	3
3. INTEGRANTES DO PROJETO E SUAS FUNÇÕES.....	4
4. PROGRAMA E BIBLIOTECAS IMPLEMENTADAS.....	4
5. RESULTADOS EXPERIMENTAIS.....	5
6. CONCLUSÃO.....	6
7. BIBLIOGRAFIA.....	7
8. ANEXOS.....	8

1. INTRODUÇÃO

O entendimento da topologia e geografia de uma região é de grande importância em áreas como geologia e engenharia, principalmente dentro do âmbito de planejamento, pois permite a antecipação de eventuais dificuldades no estudo/obra em questão.

Um mapeamento em curvas de nível, por exemplo, é imprescindível como ferramenta de representação da região a ser trabalhada, permitindo uma pré-visualização e auxiliando no planejamento, porém, não enfatiza a dimensão de profundidade de forma precisa, nem determina a existência de possíveis obstáculos, como pedras e lençóis freáticos, havendo a necessidade de se consultar outras formas de registro.

Vários dados geológicos podem ser renderizados como mapas de contorno em geologia estrutural, sedimentologia, estratografia e geologia econômica. São usados para mostrar o que há por baixo da superfície do solo, de estratos geológicos, falhas na superfície e não-conformidades [1]. A base para se compilar mapas geológicos é geralmente um mapa topográfico no qual a diversidade do terreno é expressa pelas linhas de contorno, linhas que agrupam todos os pontos de mesma altura acima do nível do mar que, quando formam uma curva fechada, podem ser representados por um símbolo indicando o tipo de material, de acordo com [2].

O projeto visa justamente enriquecer a visualização a partir de poucos dados, e possibilitar o estudo de objetos (canos, tubos, túneis, etc) que já existam no terreno ou que serão colocados como parte de uma obra, possibilitando a verificação de modificações que se fazem necessárias no terreno para que o objeto possa ser colocado no local.

2. OBJETIVOS

A motivação deste projeto é, dado um terreno no formato pgm como entrada, gerar uma visualização em três dimensões deste terreno e de seus aspectos mais importantes para o estudo/obra em questão, visando suprir a deficiência apresentada pelas curvas de nível. As camadas compreenderão elementos de interesse no terreno, como lençóis freáticos, tubulações, rochas, linhas de metrô, etc e irão se apresentar de maneira sobreposta umas as outras de forma a ser possível, por meio de efeitos de transparência, visualizar apenas as camadas desejáveis.

Além disso, será possível colocar a textura (imagem em formato ppm) que represente a região superficial do terreno.

3. INTEGRANTES DO PROJETO E SUAS FUNÇÕES

Como metodologia foi adotado o uso de cartões de tarefas, que consiste em exaurir todas as funções que se espera haver no programa e escrever essas funções em cartões, sendo que cada integrante ficará responsável por um certo número de cartões e, portanto, responsável por implementar a função descrita nesse cartão.

Além do *website* do projeto [3], onde é possível ver *screenshots*, também criamos um grupo de discussão no Google Groups, para centralizar a troca de mensagens a respeito do projeto e combinar datas para reuniões.

Felipe Augusto A. Dias – Codificação das funções relacionadas às entradas requeridas pelo programa.

Franklin de Oliveira – Codificação da interface do usuário e de visualização.

Giseli de A. Ramos – Documentação e codificação das funções relacionadas aos elementos do terreno e funções de visualização dos elementos.

João Pedro Kerr Catunda – Manutenção do código (servidor SVN), coordenação das atividades e codificação das funções relacionadas à renderização do terreno.

Vivian Cury – Codificação dos componentes do terreno.

4. PROGRAMA E BIBLIOTECAS IMPLEMENTADAS

O programa está sendo desenvolvido na linguagem de programação C, e utiliza as bibliotecas do OpenGL/GLUT/GLU [4][5][6]. Para a utilização do *mousewheel*, está sendo utilizada a biblioteca *freeglut*.

As funcionalidades apresentadas pelo programa são:

- Renderização do terreno e subsolo em modo *wireframe* e/ou sólido;
- Uso de textura;
- Rotação do terreno ao redor de qualquer um dos eixos x, y ou z;
- Zoom pelo teclado ou via *mousewheel*;
- Carregar um arquivo pgm de terreno ou ppm de textura através do menu;
- Translação do terreno pelo teclado;
- Inserção de objetos simples como pedras e canos.

No caso da renderização do terreno/subsolo, a imagem de entrada é um arquivo no formato

pgm. O programa então o converte em uma matriz numérica, onde cada célula ij representa a posição no plano xy e o valor da célula representa a altura no eixo z . Dessa forma, a matriz pode ser renderizada em uma imagem tridimensional.

Como forma de manipulação da imagem 3D, é possível usar a função de *zoom* no terreno através do *mousewheel*, usando, para tanto, a biblioteca *freeglut*. Além disso, o terreno pode ser rotacionado em torno de cada um dos eixos (x , y ou z), mas apenas em um deles de cada vez. O programa ainda consegue realizar a traslação no terreno, permitindo centralizá-lo na tela ou focar em outros pontos.

5. RESULTADOS EXPERIMENTAIS

Como entrada, o programa recebe um arquivo de texto, onde cada linha codifica para uma certa informação, dependendo do caractere inicial. São eles:

Tabela 1 – Significado dos caracteres iniciais no arquivo de entrada e estrutura da codificação.

Caractere inicial	Significado	Caracteres posteriores	Significado
# (hash)	Comentário (linha ignorada pelo programa).	-	-
L	Dimensões do terreno.	x y	x e y indicam as dimensões do terreno a ser lido.
n	Número de elementos de cada tipo.	c p t	c, p e t indicam, respectivamente, o número de canos, pedras e superfícies (terreno e subsolos) a serem lidos.
s	Indicação de superfície.	rel text	rel é o caminho para o arquivo pgm de relevo e text é o caminho para o respectivo arquivo ppm de textura.
c	Indicação de cano.	x0 y0 z0 x1 y1 z1 r	x0, y0 e z0 é o ponto inicial do cano e x1, y1 e z1 o ponto final. Já o parâmetro r representa o raio do cano.
p	Indicação de pedra.	x0 y0 z0 r dx dy dz	x0, y0 e z0 é o ponto onde a pedra se encontra e d, o diâmetro da pedra. Já dx, dy e dz, representam parâmetros de deformação nos respectivos eixos, dado à esfera uma forma mais ovalada.
f	Fim de arquivo.	-	Indica que o fim do arquivo foi atingido.

Um exemplo de entrada encontra-se na seção 8. Anexos.

O formato pgm foi adotado por definitivo principalmente devido aos problemas encontrados para se fazer a leitura de arquivos em outros formatos. No entanto, se por um lado a restrição de 8 bits de níveis de cinza desse tipo de formato consiste em perda de resolução e impossibilidade de se trabalhar com terrenos que apresentem extremos de elevação, por outro, o uso desse formato facilitou a leitura do arquivo, permitindo que o foco fosse voltado para outros elementos mais relevantes do programa. Além disso, grande parte das obras e estudos que necessitam da visualização de terrenos são feitas em áreas relativamente pequenas e dificilmente apresentam valores extremos de elevação [8].

Outra forma de entrada para o terreno que havia sido cogitada foi a busca de valores de elevação via WEB através da latitude e longitude de pontos na superfície terrestre. Embora esse tipo de serviço exista [7], ele é um dos poucos desse tipo, além de ser lento e apresentar imprecisões. Mesmo outros serviços via WEB do ramo, além de também serem lentos, proporcionam apenas arquivos com formatos muito diferentes do usual, sendo um impedimento à portabilidade.

Como pode ser visto pela Tabela 1, tanto o terreno (superfície) quanto as camadas abaixo dele (subsolo) são lidas e renderizadas da mesma forma, permitindo que todas as camadas possam ser trabalhadas da mesma forma.

As primitivas cano e pedra são representadas por formas simples, respectivamente cilindro e esfera. Para se construir sistemas de tubulação, basta especificar o ponto inicial de um cano como tendo o mesmo valor do ponto final de outro.

A visualização do terreno e do subsolo pode ser feita no modo *wireframe* e/ou sólido, ou seja, o terreno pode ser mostrado como uma malha vazada (*wireframe*), como uma superfície sólida (sólido) ou como uma superfície sólida juntamente com a malha (*wireframe* e sólido). Além disso, também é possível texturizar o terreno passando um arquivo de textura no formato ppm quando da entrada de uma superfície, assim como indicado na Tabela 1. A visualização gerada pelo OpenGL é feita através do algoritmo de *z-buffer*.

A iluminação presente no terreno 3D é apenas de luz ambiente e, portanto, envolve o algoritmo de radiosidade.

6. CONCLUSÃO

Embora nem todas as propostas que à princípio foram imaginadas para o programa tenham conseguido ser implementadas, os objetivos estabelecidos foram alcançados. O programa apresenta as funcionalidades básicas que eram esperadas do mesmo, como a visualização tridimensional de um terreno e a manipulação dessa visualização através de *zoom* e rotação. Além disso, conseguiu-se

manter a entrada do programa em uma forma simplificada e de fácil entendimento para o usuário, sem que este tenha que se preocupar muito com a parte interna do programa.

Durante o desenvolvimento do projeto, a computação gráfica se mostrou de fato uma ferramenta poderosa para a ilustração das mais diversas questões e seu aprendizado, não apenas no âmbito da programação, mas também no campo teórico, se mostrou imprescindível para a formulação de soluções que demandem a utilização de recursos gráficos.

7. BIBLIOGRAFIA

- [1] Contour lines, http://en.wikipedia.org/wiki/Contour_line, acessado em 22/04/2010.
- [2] Simpson, Brian. Geological Maps. Pergamon Press, 1985.
- [3] Página Web do projeto GeoGrid:
http://linux.ime.usp.br/~jpkc/cg/Computacao_Grafica/GeoGrid.html.
- [4] Jackie Neider, Tom Davis, Mason Woo. OpenGL Programming Guide – The Official Guide to Learning OpenGL, Release 1. 1995. Addison Wesley.
- [5] Computação Gráfica - Biblioteca Gráfica OpenGL,
<http://www.inf.pucrs.br/~pinho/CG/Aulas/OpenGL/OpenGL.html>, acessado em 20/04/2010.
- [6] Introdução ao OpenGL, <http://www.inf.pucrs.br/~manssour/OpenGL/Tutorial.html>, acessado em 20/04/2010.
- [7] Dados de elevação de um ponto na superfície da Terra, <http://code.google.com/intl/pt-BR/apis/maps/documentation/elevation/>, acessado em 14/05/2010.
- [8] Greg Magarshak. Theory & Practice - Issue 05 - Landscapes,
http://www.flipcode.com/archives/Theory_Practice-Issue_05_Landscapes.shtml, acessado em 01/04/2010.
- [9] Jonas Gomes and Luiz Velho. "Fundamentos da Computação Gráfica". IMPA, 2003.
- [10] Paul Martz. Generating Random Fractal Terrain,
<http://www.gameprogrammer.com/fractal.html>, acessado em 20/04/2010.

8. ANEXOS

8.1. Exemplo de entrada:

```
#Comentários começam com #
#L nos dá as dimensões do terreno.
#L x y
L 512 512
#n determina quantos canos, pedras e quantas superfícies temos
#n canos pedras superficies
n 1 0 2
#s nos dá um superficie e sua textura
#s relevo textura
s data/agua.pgm data/agua.ppm
s data/terreno.pgm data/terreno.ppm
#c nos dá um cano
#c x0 y0 z0 x1 y1 z1 raio
c 12 12 12 36 36 36 2
#f indica fim de arquivo
f
```