

Uma Ferramenta Computacional para geração de Modelos Digitais de Terrenos com visualização em VRML

Antonio de Paula Pedrosa

Universidade Severino Sombra-USS Vassouras-RJ, Brasil
tonywebdevelop@hotmail.com

Carlos Vitor de Alencar Carvalho

Universidade Severino Sombra-USS Vassouras-RJ, Brasil,
Centro Universitário de Volta Redonda-UniFOA, Volta Redonda-RJ,
Brasil e Instituto Superior de Tecnologia -IST/FAETEC,
Paracambi-RJ, Brasil - cvitorc@gmail.com

Resumo: *Este artigo apresenta o desenvolvimento de uma aplicação computacional para geração de Modelos Digitais de Terrenos (MDT). Neste trabalho é utilizado o método denominado de Inverso do quadrado da distância. Esse método de interpolação, comumente é utilizado em sistemas de informação geográfica (SIG) para geração de mapas a partir de dados pontuais. No desenvolvimento foi utilizada a linguagem Java e a visualização da superfície gerada foi feita por meio da linguagem VRML. Como exemplo da metodologia adotada, foi gerado um MDT de parte da região da cidade de Vassouras, Rio de Janeiro.*

Palavras-Chave: *Modelo Digital de Terreno, Linguagem Java e VRML*

A Computational Tool for generation of Digital Terrain Models with visualization in VRML

Abstract: *This paper presents the development of a computer application for generation of Digital Terrain Models (DTM). This work is being used a method called the inverse square distance. This method of interpolation is commonly used in geographic information systems (GIS) to generate maps from point data. The development was done using the Java language and visualization of the surface*

was generated by VRML. As an example of the methodology used, was generated from a DTM of the region of the city of Vassouras, Rio de Janeiro.

Keywords: *Digital Terrain Model. VRML and Java Languag.*

1. Introdução

A computação nos dias atuais tem sido utilizada nas mais diversas áreas. Numa delas há uma demanda muito grande: a de aplicação para solução de problemas de engenharia. Uma das aplicações na engenharia, em que a componente computacional é importante é na geração de superfícies para visualização de uma determinada característica de estudo. Geralmente, as superfícies são geradas de dados que não se conhecem e sua distribuição em todos os seus pontos, isto é, os dados são gerados mediante modelagem matemática de forma analítica ou com base nos dados pontuais coletados. Essas superfícies são chamadas de Modelos Digitais de Terrenos (MDT). A superfície gerada é, em geral, contínua e a característica representada pode ser de várias áreas de conhecimento, segundo Burrough, (1986) *apud* Medina (2007). Alguns usos do MDT são:

- Armazenamento de dados de altimetria para mapas topográficos;
- Análises de corte-aterro para projeto de estradas e barragens;
- Elaboração de mapas de declividade e exposição para apoio à análise de geomorfologia e erodibilidade;
- Análise de variáveis geofísicas e geoquímicas;
- Apresentação tridimensional (em combinação com outras variáveis).

O processo de geração de um MDT pode ser resumido na figura 1. Fundamentalmente, existem três procedimentos: aquisição dos dados, geração do *grid* e visualização da superfície gerada.

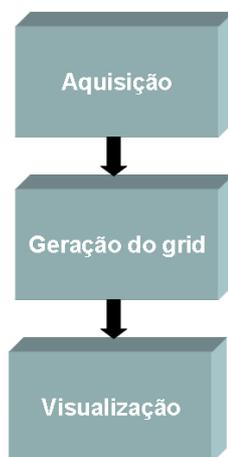


Figura 1 – processo de geração de um MDT

Este trabalho apresenta um sistema computacional para geração de um MDT desenvolvido em linguagem Java, em que a visualização foi feita por meio da linguagem VRML. O algoritmo utilizado para a geração da superfície foi o Inverso do Quadrado da Distância (IQD). Trata-se de um método de interpolação local, determinístico e exato, comumente utilizado em Sistemas de Informação Geográfica (SIG) para geração de mapas a partir de dados pontuais. (Landim 2000).

O restante deste artigo está dividido da seguinte forma: A seção 2 apresenta detalhes do algoritmo utilizado para a geração do *grid*. Na seção 3 são apresentados os detalhes da metodologia adotada. Na seção 4 os resultados obtidos com a geração do MDT de parte da região da cidade de Vassouras, Rio de Janeiro. Em seguida, são apresentadas as considerações finais e conclusões do trabalho.

2. Algoritmo para a geração do *grid*

Para a geração de um MDT, quando não se conhecem todas as informações da região a ser modelada, é necessária a definição de um modelo matemático. Existem diversos desses modelos em que se destacam dois grupos: funções globais e locais. Nos métodos de funções globais são considerados todos os pontos da área, que permitem interpolar o valor da função em qualquer ponto dentro do domínio dos dados originais. A adição ou remoção de um valor terá consequências no domínio de definição da função. Como exemplos têm-se: polinômios e equações multiquádricas, que nos métodos de funções locais são definidas para porções do mapa. A alteração de um valor afetará localmente os pontos que lhes são próximos. Como exemplos dessa metodologia tem-se: triangulação e inverso da potência das distâncias. (Landim 2000).

Como dito anteriormente, neste artigo foi método utilizado para geração do MDT, o método chamado Inverso do Quadrado da Distância (IQD). Trata-se de interpolação local, determinístico e exato, comumente utilizados em Sistemas de Informação Geográfica (SIG) para a geração de mapas, a partir de dados pontuais. (Landim 2000). Neste método, as cotas do *grid* são atualizadas pelas médias ponderadas das amostras. Os pesos devem levar em conta a pouca influência dos pontos mais distantes na determinação das grandezas desconhecidas, ou seja, o peso é utilizado como sendo o inverso da distância do ponto (x, y) de que se quer obter o valor da subsidência a uma das amostras. O valor estimado da grandeza é dado pela equação (1). Para os pontos pertencentes ao conjunto de amostras, utiliza-se a equação (2). Isso evita erros de interpolação para valores pré-existentes. A Figura 2 apresenta de forma esquemática o *grid* que será gerado e os pontos conhecidos e interpolados (Carvalho 2002).

$$z(x, y) = \frac{\sum_{i=1}^n \frac{z_i}{h_{ij}^\beta}}{\sum_{i=1}^n \frac{1}{h_{ij}^\beta}} \quad (1)$$

$$h_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (2)$$

$$z(x, y) = z_i \quad (3)$$

Onde:

$z(x, y)$ - valor interpolado para o nó do *grid*, onde x e y são as posições cartesianas do ponto z .

z_i - valor conhecido da amostra i ;

h_{ij} - distância entre o nó do *grid* e o ponto da amostra i . É calculado pela eq. (2);

β - expoente de ponderação que neste caso é igual a 2.

n - número de amostras utilizadas para interpolar cada nó do *grid*.

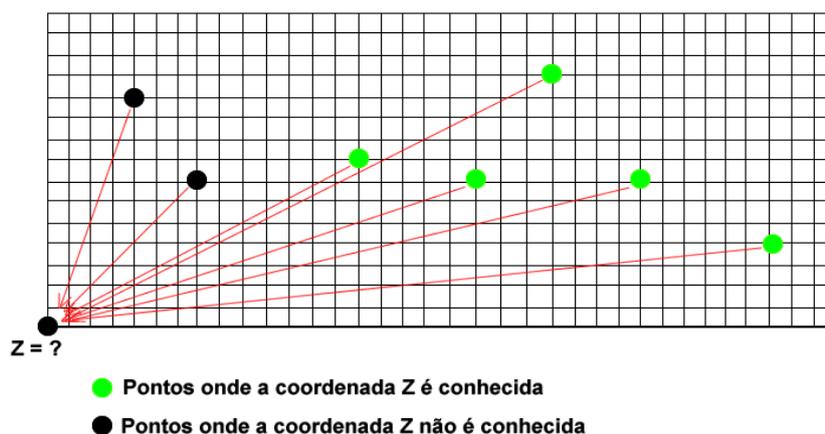


Figura 2 – esquema do *grid* gerado indicando pontos conhecidos e pontos que serão interpolados

3. Metodologia

A seguinte metodologia foi adotada para a aquisição, geração e visualização do MDT de parte da região da cidade de Vassouras, Rio de Janeiro.

3.1. Aquisição

Inicialmente por meio do *google earth* (GoogleEarth 2010) foi definida uma região da cidade de Vassouras, de forma a contemplar os principais pontos da cidade. Principalmente, foram marcados quatro pontos principais delimitando a região a ser modelada: as distâncias entre as marcações que delimitam a região estão equidistante de 3 quilômetros (Tabela 1 e Figura 3). No modelo, o *grid* foi dividido em 10 (dez) partes com 300 (trezentos) metros cada uma, num total de 3 quilômetros de um ponto ao outro. Também foram selecionados 10 (dez) pontos principais da cidade. Os pontos escolhidos foram: Universidade Severino Sombra, Igreja Matriz, Câmara Municipal, Ginásio Esportivo-Sombrão, Shopping BR - 393, Hospital Universitário, Monumento Centenário, Memorial Manoel Congo, Morro da Torre e Fórum (Tabela 2 e Tabela 3).

Tabela 1 – distância entre as marcações que delimitam a região a ser modelada

Marcação A	Marcação B	Distância
Marc. 1	Marc. 2	3 km
Marc. 1	Marc. 3	3 km
Marc. 2	Marc. 4	3 km
Marc. 3	Marc. 4	3 km



Figura 3 – Região a ser modelada indicando as marcações 1, 2, 3 e 4.

Tabela 2 – Coordenadas dos pontos da cidade que foram selecionados e suas respectivas elevações.

Nº	Pontos Marcados	Elevações
1	Universidade Severino Sombra	431m
2	Igreja Matriz	441m
3	Câmara Municipal	428m
4	Sombrão	439m
5	Shopping BR-393	415m
6	Hospital Universitário Sul Fluminense	434m
7	Monumento Centenário	464m
8	Memorial Manoel Congo	471m
9	Morro da Torre	592m
10	Fórum	415m



Figura 4 – Região a ser modelada indicando os pontos da cidade que foram selecionados.

Tabela 3 – Coordenadas dos pontos da cidade que foram selecionados e suas respectivas posições na direção x e y.

Pontos Marcados	X	Z
Monumento Centenário	620	312
Shopping BR-393	736	781
Universidade Severino Sombra	1549	855
Igreja Matriz	1772	775
Câmara Municipal	1829	934
Memorial Manoel Congo	1986	554
Fórum	2008	1894
Sombrão	2240	1411
Morro da Torre	2510	727
Hospital Universitário Sul Fluminense	2634	2139

3.2 Geração e Visualização

O sistema, após gerar a superfície, exporta um arquivo no formato da linguagem VRML para visualização. A Figura 4 ilustra o procedimento de exportação e importação do arquivo gerado. A geração do *grid* do MDT por meio do método do IQD foi implementada em linguagem Java. Parte do código desenvolvido pode ser visualizado no quadro 1.

A linguagem VRML (Vrml 2010) foi utilizada por ser um padrão para desenvolvimento de aplicações de realidade virtual multiusuário na *Internet*. (Portela 2010). Trata-se de uma linguagem independente de plataforma que permite a criação de ambientes virtuais, por onde se pode passear, visualizar objetos por ângulos diferentes e até com eles interagir. A linguagem trabalha com geometria 3D mediante de algumas primitivas básicas (cubo, cone, cilindro e esfera) e suporta transformações (rotação, translação, escala), texturas, luz e sombreamento.

Para navegar em mundos virtuais criados com a linguagem é necessário usar *browsers* que suportem VRML. Existem muitos *browsers* disponíveis que suportam diretamente a linguagem. Outros *browsers* que não suportam necessitam de software adicional (*plug-in*) (Portela 2010).

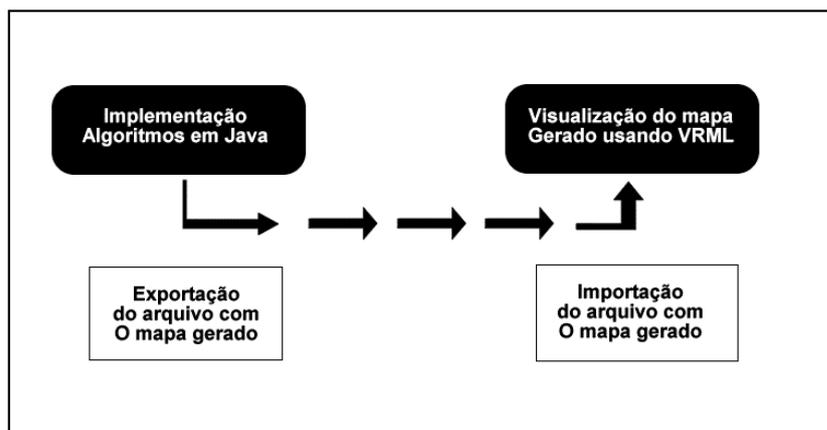


Figura 4 – esquema do processo para a geração e visualização do MDT

Quadro 1 – parte principal do código implementado.

```
// Realizando o calculo das elelações do GRID
double soma1 = 0;
double soma2 = 0;

for (int lin = 0; lin < mat.length; lin++) {
    for (int col = 0; col < mat[lin].length; col++) {
        xgrid = mat[lin][col].getX();
        zgrid = mat[lin][col].getZ();
        for (int k = 0; k < ndivX * ndivY; k++) {

            xconhecido = pontosconhecidos.get(k).getX();
            zconhecido = pontosconhecidos.get(k).getZ();
            H = Math.sqrt((xconhecido - xgrid) * (xconhecido - xgrid) - (zconhecido - zgrid) * (zconhecido - zgrid));
            if (H > tol) {
                Hexp = Math.pow(H, 4);
                soma1 = soma1 + (pontosconhecidos.get(k).getY() / Hexp);
                soma2 = soma2 + (1 / Hexp);
                ygrid = soma1 / soma2;
            }
        }
        mat[lin][col].setY(ygrid);
    }
}
```

4. Resultados

De início, procurou-se verificar o potencial da linguagem VRML no que diz respeito à visualização, pela criação de um terreno com dados randômicos e sem textura (Figura 5) e com textura (Figura 6).

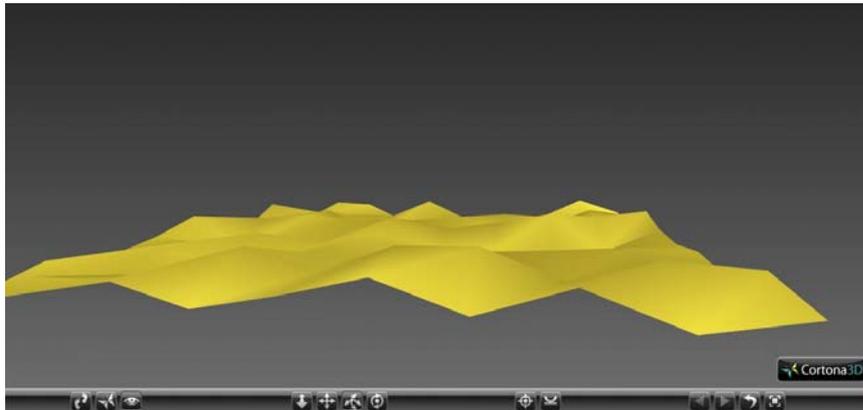


Figura 5 – Visualização de uma superfície com dados randômicos

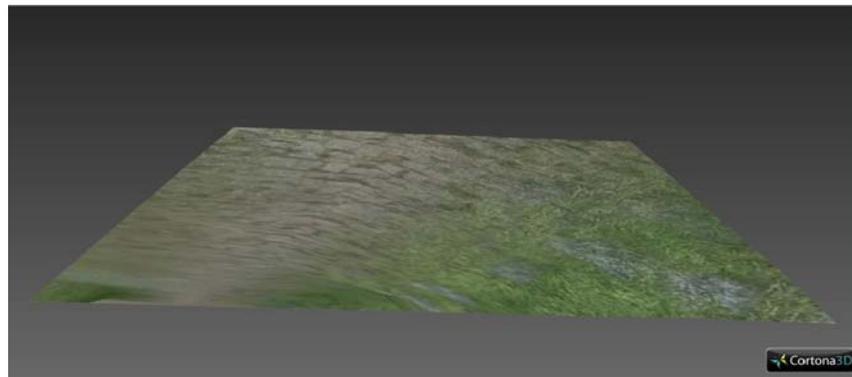


Figura 6 – Terreno visualizado com textura

Com a implementação do método IQD foi possível chegar a um ponto aproximado do terreno MDT, com as elevações existentes. Nessa primeira exibição não há nenhum tipo de textura, como podemos ver na figura 7.

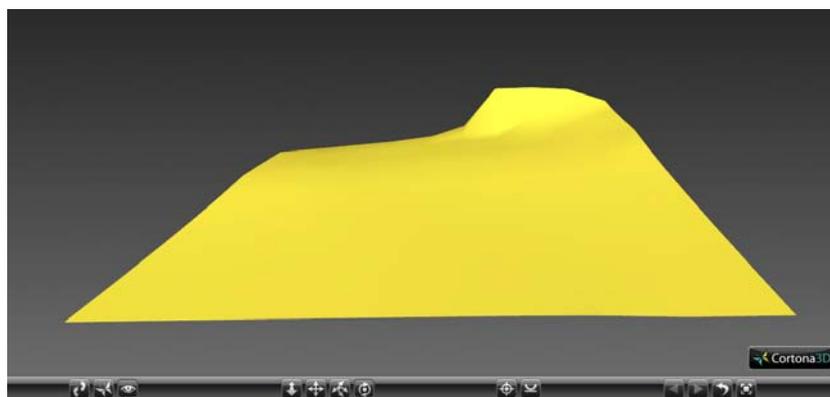


Figura 7 – 1º modelo de terreno gerado a partir do método IQD, visualizado ainda sem textura.

Na próxima imagem (Figura 8) vemos o mesmo terreno, agora contendo a textura.



Figura 8 – 1º modelo de terreno gerado a partir do método IQD, visualizado ainda com textura - Imagem visualizada pelo usuário

Com a funcionalidade do *plugin* adquirido em Cortona3D (2010), podemos alcançar o terreno de diferentes ângulos, com maior ou menor visibilidade, como nos mostram as figuras a seguir (Figura 9).

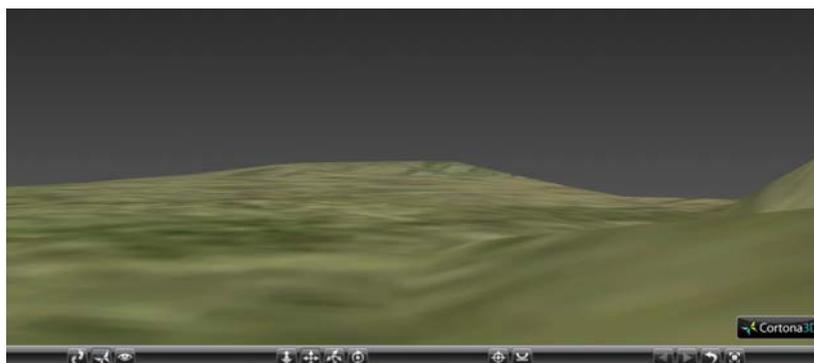


Figura 9 – Maior visibilidade do terreno mediante de aproximação (Ângulo 1).

Nas figuras 10 e 11 podemos visualizar, em uma parte do terreno, suas elevações com mais detalhes.



Figura 10 – Maior visibilidade do terreno mediante de aproximação (Ângulo 2).



Figura 11 – Maior visibilidade do terreno mediante de aproximação (Ângulo 3).

5. Considerações Finais

Acreditamos que o primeiro objetivo desse projeto foi cumprido com o desenvolvimento de uma aplicação computacional para geração de um MDT. A ideia é aperfeiçoar o modelo gerado, para que possa ter objetos que identifiquem os pontos da cidade, selecionados conforme indicado na Tabela 2, que atribui ao terreno características de uma cidade em 3D.

Agradecimentos

O primeiro autor agradece à FAPERJ pelo apoio financeiro de Bolsa de Iniciação Científica. O segundo agradece ao CNPq pelo apoio financeiro através da Bolsa de Produtividade em Desenvolvimento Tecnológico e Extensão Inovadora – DT.

Referências

- Burrough, P.A.; McDonnell, R.A. *Principles of geographical information Systems*. Oxford University Press. 1986. 333p.
- Carvalho C. V. A., “*Simulação de Transporte e Deposição de Sedimentos Siliciclásticos em Ambientes de Plataforma, Talude e Bacia*”, Tese de doutorado, Departamento de Engenharia Civil/Tecgraf, PUC-Rio, 2002.
- Cortona3d. Cortona3D Viewer – *User’s Guide*. June 2009. Disponível em: <http://82.165.194.128/cortona3d/files/a2/a2607c21-b8b6-4abf-8f5d-e3abb2a67649.pdf>. Último acesso em 01 de Abril de 2010.
- Cortona3d. *Cortona3D Viewer*. Disponível em: <http://www.cortona3d.com/Products/Cortona-3D-Viewer.aspx>. Último acesso em 01 de Abril de 2010.
- Google Earth. Disponível em: <http://earth.google.com/>. Último acesso em 10 de Setembro de 2010.
- Lamdim, P. M., B., “*Texto Didático 2 – Introdução aos Métodos de Estimatição espacial para confecção de mapas*”, UNESP/Rio Claro, 2000.
- Medina, A. S., Medina, S. da S. S. *A Representação da superfície topográfica através de modelos digitais de terrenos*. VII International Conference on Graphics Engineering for Arts and Design XVIII Simpósio Nacional de Geometria Descritiva e Desenho Técnico, Curitiba, Paraná, Brasil, 2007.
- Portela, F. C. *Tutorial VRML*. Disponível em: <http://www.franklin.pro.br/apostilas/apostilaVRML.pdf>. Último acesso em 20 de Setembro de 2010.
- Vrml. *VRML Interactive Tutorial*. Disponível em: <http://sim.di.uminho.pt/vrmltut/>. Último acesso em 04 de Abril de 2010.