

BCG - Boletim de Ciências Geodésicas - On-Line version, ISSN 1982-2170  
<http://dx.doi.org/10.1590/S1982-21702014000100008>

---

# DESENVOLVIMENTO DE UMA PLATAFORMA DE SOFTWARE PARA A MODELAGEM DIGITAL DE TERRENOS BASEADA EM TIN

*Developing an opensource software system for the digital terrain model based on TIN*

F. F. OLIVEIRA<sup>1</sup>  
M. A. PITERI<sup>2</sup>  
M. MENEGUETTE<sup>2</sup>

Universidade Estadual Paulista-UNESP  
Faculdade de Ciências e Tecnologia-FCT

<sup>1</sup>Departamento de Cartografia

<sup>2</sup>Departamento. de Matemática, Estatística e Computação

Rua Roberto Simonsen, 305

1906-900 – Presidente Prudente - SP

ffoplay@gmail.com ; (piteri, messias)@fct.unesp.br

## RESUMO

Superfícies topográficas podem ser representadas com um bom grau de precisão por meio de mapas. No entanto, estes nem sempre são as melhores ferramentas para a compreensão de relevos mais complexos. Nesse sentido, a maior contribuição desse trabalho é a especificação e implementação da arquitetura de uma plataforma de software *opensource* voltado para a representação de modelos digitais de terrenos baseado em TIN (Triangular Irregular Network), segundo os paradigmas da programação orientada a objetos e programação genérica e que possibilitaram a integração de várias ferramentas de linguagem aberta (*opensource*), tais como GDAL, OGR, OpenGL, OpenSceneGraph e Qt. Além disso, o núcleo de representação da arquitetura tem a capacidade de trabalhar com múltiplas estruturas de dados topológicas que permitem extrair em tempo constante, todas as relações de conectividade entre as entidades vértices, arestas e faces, presentes numa subdivisão planar triangular, facilitando enormemente a implementação de aplicações de tempo

real. Esta capacidade é muito importante, por exemplo, na utilização de dados de levantamentos a laser (Lidar, ALS, TLS), possibilitando a geração de modelos de malhas triangulares da ordem de milhões de pontos.

**Palavras-chave:** Visualização; Triangulação de Delaunay; Estruturas de Dados Topológicas; MDT; Sistemas Opensource.

### ABSTRACT

Topographical surfaces can be represented with a good degree of accuracy by means of maps. However these are not always the best tools for the understanding of more complex reliefs. In this sense, the greatest contribution of this work is to specify and to implement the architecture of an *opensource* software system capable of representing TIN (Triangular Irregular Network) based digital terrain models. The system implementation follows the object oriented programming and generic paradigms enabling the integration of various opensource tools such as GDAL, OGR, OpenGL, OpenSceneGraph and Qt. Furthermore, the representation core of the system has the ability to work with multiple topological data structures from which can be extracted, in constant time, all the connectivity relations between the entities vertices, edges and faces existing in a planar triangulation what helps enormously the implementation for real time applications. This is an important capability, for example, in the use of laser survey data (Lidar, ALS, TLS), allowing for the generation of triangular mesh models in the order of millions of points.

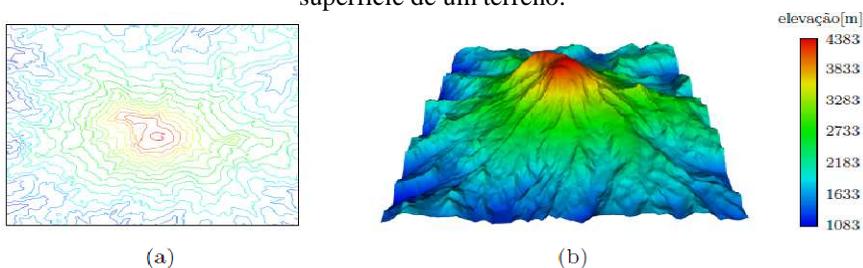
**Keywords:** Visualization; Delaunay Triangulation; Topological Data Structures; DTM; Opensource Systems.

## 1. INTRODUÇÃO

A representação da superfície terrestre ou parte dela de forma conveniente e com determinado grau de precisão tem sido um constante desafio para a comunidade científica e os mapas correspondem ao meio mais utilizado para esse fim. Os mapas modernos são elaborados com altas exigências de acurácia possibilitando que informações de natureza espacial (ex: posição, orientação, medidas de distância, área e volume) possam ser extraídas dos mesmos com precisão e segurança (LI; ZHU; GOLD, 2005). Em particular, mapas de curvas de nível são os meios mais usuais de representação estática e bidimensional de uma superfície topográfica. Nesse tipo de mapa, a superfície do terreno é descrita por linhas com valores de elevação constante (isolinhas).

Dentre os inconvenientes dessa forma de representação destacam-se a impossibilidade de se representar anomalias do terreno entre as isolinhas sem o uso de métodos de interpolação (Figura 1a) e a retirada de informações tridimensionais não é intuitiva, cabendo aos usuários a reconstrução e visualização mental da realidade 3D neste representada (Figura 1b).

Figura 1 - Representação bidimensional (a) e perspectiva tridimensional (b) da superfície de um terreno.



O conhecimento sobre a topografia de um terreno é de grande importância para a área de Ciências da Terra. A utilização de Modelos Digital de Terreno (MDTs) em combinação com diferentes dados espaciais pode resultar numa importante base de dados para análises topográficas em diferentes áreas como: Sensoriamento Remoto, Ecologia, Hidrologia e Geomorfologia (EL-SHEIMY; VALEO; HABIB, 2005; BARBOSA, 1999). Entretanto, o emprego de MDTs não é restrito à área de Ciências da Terra, podendo ser necessário ou até mesmo um pré-requisito para muitas aplicações de engenharia civil, militar e também em áreas industriais como telecomunicações, navegação, transporte e planejamento de infraestrutura (SULEBAK, 2000).

Existe uma quantidade relativamente razoável de aplicações comerciais que fazem uso ou contêm ferramentas para a manipulação de MDTs. Entre eles podem ser citados o *Topograph*, *Idrisi*, *Envi*, *ArcGIS*, *LPS*, *SOCSET* e o *SURFER*. Contudo, a maior parte dessas aplicações é voltada para Sistema de Informação Geográfica (SIG), possuindo apenas módulos para a modelagem de terrenos, não sendo específicas para esse fim. Outro fator importante é a ausência de programas livres nessa área. Apesar da existência de programas consistentes como *Spring*, *Landserf* ou *Topocal*, esses sofrem dos mesmos problemas relacionados às aplicações de uso comercial, não existindo programas exclusivos e genéricos o suficiente para o uso em um amplo conjunto de problemas relacionados à modelagem digital de terrenos.

A partir das considerações anteriores, o objetivo central deste trabalho é discutir a especificação e implementação da arquitetura de uma plataforma de software específica para a modelagem e manipulação de modelos digitais de terrenos baseada em TIN, construída sobre uma estrutura de dados topológica robusta, eficiente e que sirva como base para o desenvolvimento de aplicações em diferentes áreas do conhecimento. De modo a ilustrar a viabilidade da proposta, inúmeras funcionalidades foram desenvolvidas e já se encontram operacionais, o que reforça a abordagem e as metodologias utilizadas.

## 2. MODELAGEM DIGITAL DE TERRENOS

No contexto do presente trabalho, o termo MDT será utilizado para a descrição de um modelo de representação de relevo que pode conter dados de elevação e elementos geográficos (ou feições SIG), como rios, estradas, linhas de quebra, etc; visto que no sistema proposto é permitida a associação de outras informações ao modelo (não apenas valores de elevação).

De uma perspectiva geral, a problemática da Modelagem Digital de Terreno pode ser dividida em três grandes áreas de pesquisa, a saber: *Aquisição, Manipulação e Aplicações*.

Basicamente, a etapa de *aquisição* compreende a obtenção de um conjunto de amostras sobre a superfície do terreno e serve de base para todas as operações que poderão ser efetuadas sobre o MDT. Os dados para elaboração dos modelos podem ser originados a partir de mapas existentes, técnicas fotogramétricas, levantamentos terrestres ou por meio de outras tecnologias como *Radio Detecting and Ranging (RADAR)* e *Light Amplification by Stimulated Emission of Radiation (LASER)*.

Por sua vez, a fase de *manipulação* está relacionada essencialmente com a construção e a representação computacional da superfície de um terreno, isto é, com a estrutura de dados combinatória e topológica que será construída a partir da amostragem realizada na fase anterior. O modelo de dados usado por um MDT deve permitir um conjunto de operações sobre o mesmo, tais como modificações, refinamentos e procedimentos de conversão entre diferentes estruturas de dados que podem ser utilizadas para representá-lo, gerando as informações necessárias a uma dada aplicação (EL-SHEIMY; VALEO; HABIB, 2005).

As *aplicações* são procedimentos de análise executados sobre o MDT como testes de visibilidade entre pontos na superfície do terreno, cálculos de distância, área e volume, geração de mapas de declividades, visualização do modelo em perspectiva, perfis e seções transversais, entre outras. O presente trabalho se limita a etapa de modelagem relativa à manipulação computacional de MDTs.

Dentre os diversos métodos existentes que podem ser utilizados para obter uma representação de superfícies em um formato digital (MDT) é possível destacar grades regulares (também conhecidas como matriz de elevação ou GRID) e malha triangular irregular (TIN) (LI; ZHU; GOLD, 2005).

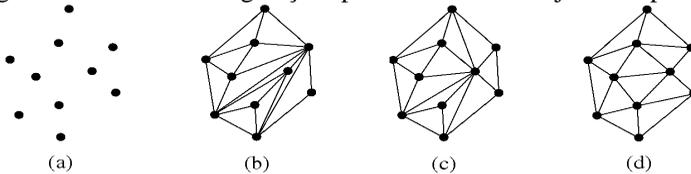
No modelo de dados GRID, a superfície do terreno é representada por uma estrutura de dados matricial onde cada um de seus elementos contém um valor de elevação ( $z$ ) do terreno amostrada em intervalos regulares no plano  $xy$  (EL-SHEIMY; VALEO; HABIB, 2005). Trata-se de um modelo de fácil armazenamento e manipulação computacional, uma vez que todas as relações topológicas entre os pontos amostrados sobre o terreno já se encontram armazenadas de forma implícita na estrutura de dados matricial. Em outras palavras, relações de vizinhança entre pontos no GRID podem ser encontradas sem a necessidade de um esforço de computação adicional (ABDUL- RAHMAN, 1994).

A desvantagem em utilizar esse modelo se deve ao fato da distribuição espacial dos pontos utilizados não estar relacionada com as características do

terreno modelado (PETRIE; KENNIE, 1987), podendo gerar inconvenientes como um número significativo de dados redundantes em áreas mais planas do terreno, além de não conseguir representar de forma precisa áreas com relevos mais complexos, o que acaba por restringir a representação de características do terreno que sejam menores que o espaçamento entre os pontos da malha.

Por outro lado, no modelo TIN a superfície do terreno é modelada por uma malha composta por triângulos adjacentes e não sobrepostos de diferentes tamanhos e orientação, que são computados a partir de pontos (com coordenadas  $xyz$ ) irregularmente espaçados e amostrados sobre a superfície que se pretende representar e se constituem nos vértices da triangulação, enquanto as arestas são geradas a partir da conexão entre vértices por meio de segmentos de retas. Entretanto, o problema de triangulação de um conjunto de pontos arbitrários no espaço euclidiano de qualquer dimensão não possui unicidade, conforme pode ser observado na Figura 2. Logo, entre todas as triangulações possíveis, qual delas é a mais adequada para representar a superfície de um terreno.

Figura 2 - Possíveis triangulações para um mesmo conjunto de pontos.



A triangulação de um conjunto de pontos  $V = \{p_1, p_2, p_3, \dots, p_n\} \subseteq R^2$  é denominada de Delaunay ( $TD(V)$ ), se e somente se, a circunferência que circunscreve cada um dos triângulos  $\Delta(p_i p_j p_k) \in TD(V), (1 \leq i \neq j \neq k \leq n)$  for vazia, ou seja, não contiver nenhum outro ponto de  $V$  em seu interior e é a ideal para a representação da superfície de um terreno a partir de pontos amostrados sobre o mesmo (BERG, 2008; BARBOSA, 1999). No plano, essa triangulação maximiza o menor dos ângulos entre todas as triangulações possíveis e é referenciado por Critério do ângulo MaxMin (PITERI *et al.*, 2007).

Uma consequência direta das propriedades acima é que os triângulos gerados são os mais regulares possíveis, ou seja, se aproximam do equilátero (Figura 2d), além de possuir uma base matemática teórica sólida e várias abordagens algorítmicas para sua construção (BERG, 2008; HJELLE; DAEHLEN, 2006).

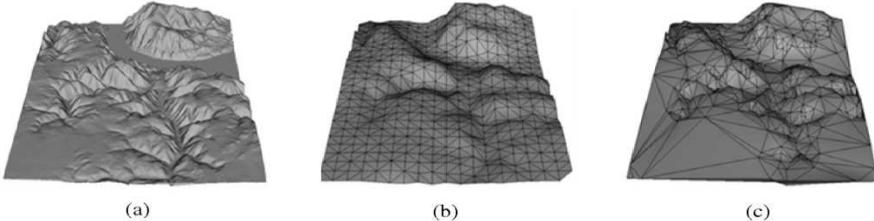
Como os pontos amostrados constituem os vértices da triangulação, o modelo TIN preserva a exata localização de cada ponto sobre terreno, melhorando assim a sua precisão (JONES; KIDNER; WARE, 1994). O modelo TIN também é adaptativo, isto é, a malha de triângulos pode ser mais densa em áreas acidentadas ou irregulares conseguindo representar melhor terrenos mais complexos e menos

densa em regiões mais planas eliminando do modelo triângulos e vértices redundantes (triângulos cujos vértices possuem uma mesma elevação) sem perdas na precisão do modelo (DANOVARO et al., 2007). Quando disponíveis, também é possível inserir nesse modelo, de forma exata, feições lineares (também chamadas de linhas de quebra) associadas a um terreno como, por exemplo, cadeias de montanhas, fundos de vale e estradas, contribuindo ainda mais para a sua precisão.

A principal desvantagem do modelo TIN em relação ao GRID é a maior complexidade na criação e manuseio da estrutura de dados utilizada para sua representação e na elaboração de procedimentos que atuem sobre a topologia do modelo com o propósito de extrair todas as nove relações de conectividade e de adjacências entre os elementos da malha triangular (vértices, arestas, faces).

A Figura 3c ilustra a representação de um terreno mostrado na Figura 3a usando o modelo TIN, enquanto na Figura 3b é possível observar as linhas horizontais e verticais associadas aos elementos quadrangulares do GRID, onde foram inseridas diagonais para facilitar o processo de renderização sólida, que utiliza elementos triangulares, independentemente de o dado ser do tipo GRID ou TIN. Para ambas as representações foram utilizadas a mesma quantidade de pontos. Entretanto observa-se que o modelo TIN concentra um número maior de pontos em regiões mais acidentadas (irregulares) e um número menor de pontos em regiões mais planas, conseguindo dessa forma, representar melhor a superfície do terreno.

Figura 3 – Modelos GRID e TIN: (a) superfície de um terreno arbitrário; (b) representação por meio do modelo GRID, e; (c) um modelo TIN.



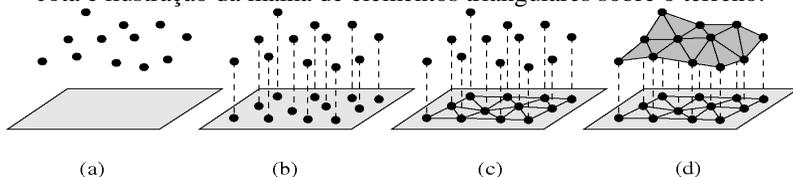
### 3. SISTEMA PROPOSTO

No sistema proposto, a modelagem da superfície do terreno é feita a partir de uma Triangulação de Delaunay. O sistema utiliza um modelo 2.5D para a representação digital de terrenos. O termo 2.5D se deve ao fato do valor de elevação (coordenada  $z$ ) ser tratado apenas como um atributo associado ao dado de posição (coordenadas  $xy$ ) sobre o terreno, não sendo permitida a associação de mais de um valor de elevação (coordenada  $z$ ) a uma mesma posição (coordenadas  $xy$ ) sobre o terreno. Nesse modelo, por exemplo, não é possível representar uma caverna, pois uma dada posição pode ter dois valores de elevação associados (um para o teto e outro para o chão). Todos os pontos utilizados também devem estar em um sistema

de coordenadas projetadas (com projeção cartográfica). A Figura 4 ilustra as etapas para a construção do modelo TIN (BERG, 2008).

Inicialmente, o sistema receberá como entrada um conjunto de pontos amostrados sobre a superfície do terreno (Figura 4a), em seguida, esse conjunto de pontos será projetado sobre o plano-xy (Figura 4b), onde então será gerada uma Triangulação de Delaunay (Figura 4c). Na sequência os pontos são levados de volta à sua posição original no espaço tridimensional (Figura 4d), preservando-se todas as informações natureza topológica (adjacências e conectividade da malha) obtidas no passo anterior (Berg, 2008; Piteri 2007).

Figura 4 - Etapas da criação de uma TIN: (a) pontos no espaço; (b) projeção sobre o plano-xy; (c) Triangulação de Delaunay dos pontos no plano-xy; (d) associação da cota e ilustração da malha de elementos triangulares sobre o terreno.



No contexto desse trabalho o termo “plataforma de software” é utilizado no sentido de que sistema proposto foi construído com o intuito de que os seus componentes chave (técnica de triangulação, estrutura de dados topológica e módulo de leitura e escrita de arquivos) possam ser utilizados como blocos construtivos para que novas funcionalidades (ex: geração de curvas de nível, cálculos de área e volume e testes de visibilidade, etc.) possam futuramente vir a ser desenvolvidas e inseridas no sistema. Já termo “sistema” se refere à aplicação desenvolvida com o intuito de mostrar a viabilidade da arquitetura proposta e das tecnologias utilizadas.

### 3.1 Ferramentas e Paradigmas de Programação Utilizados

O sistema foi construído sobre o sistema operacional *Windows* utilizando-se a linguagem de programação C/C++ (compilador *Microsoft VisualStudio*) que fornece todos os conceitos e ferramentas necessárias para a utilização dos paradigmas de programação orientada a objetos, programação genérica, além de permitir o uso da biblioteca padrão *Standard Template Library* (STL), que contém várias estruturas e componentes que facilitaram enormemente o desenvolvimento do projeto além de servir como modelo para boas práticas de programação. Padrões de projeto também foram recursos utilizados.

Considerando a grandeza e multidisciplinaridade do projeto, foram utilizadas ainda as bibliotecas e ferramentas de software *Computational Geometry Algorithms Library* (CGAL), *Geospatial Data Abstraction Library* (GDAL), *Simple Feature Library* (OGR), Qt (Q-Toolkit), *Opensource Graphics Library* (OpenGL) e

*OpenSceneGraph* (OSG). Cada uma dessas ferramentas foi escolhida levando-se em consideração desempenho, estabilidade, robustez e inserção em sua área de atuação. Vale ressaltar que essas bibliotecas de software são livres, implementadas em C/C++, suportadas em diferentes sistemas operacionais e têm se tornado padrão no mercado de desenvolvimento de software. Na sequência é realizada uma breve descrição de cada uma delas.

### 3.1.1 Biblioteca de Software CGAL

Disponibiliza um conjunto significativo de estruturas de dados e algoritmos extremamente robustos e confiáveis para o desenvolvimento de aplicações complexas na área de Geometria Computacional, capaz inclusive de trabalhar com aritmética real de precisão infinita (KETTNER; NAHER, 2004). No presente trabalho foram utilizados os algoritmos e estruturas de dados fornecidos pela CGAL ligados à criação e manutenção da Triangulação de Delaunay utilizada na representação computacional da TIN.

### 3.1.2 API de Desenvolvimento OpenGL

Sistema gráfico simples e interativo para modelagem e exibição tridimensional, rápido e independente de plataforma, que permite ao desenvolvedor de software escrever programas que acessem o *hardware* gráfico. Dentre os recursos gráficos disponíveis, podem ser destacados a renderização 3D de objetos bidimensionais (pontos, linhas e polígonos), mapeamento de superfícies com textura, modelos de iluminação, transformações geométricas (rotação, translação e escala) e de sistemas de coordenadas. O OpenGL, em conjunto com a biblioteca OSG foi utilizado na implementação das funcionalidades de visualização 3D de malhas triangulares (TIN) no sistema desenvolvido.

### 3.1.3 Biblioteca de Software OpenSceneGraph (OSG)

Permite a construção de um grafo de cena sobre o OpenGL possibilitando a montagem e gerenciamento eficiente de cenas 3D. Essa biblioteca é inteiramente escrita em C/C++ utilizando a STL e conceitos de orientação a objetos, além de implementar diferentes padrões de projetos. Usada de forma integrada ao OpenGL para visualização das TINs geradas pelo sistema.

### 3.1.4 Biblioteca de Software GDAL e OGR

Oferecem suporte à leitura e escrita de uma grande variedade de formatos de arquivos de dados espaciais *raster* e vetoriais, assim como ferramentas de linha de comando (aplicações console) para o processamento e conversão entre esses formatos. A GDAL dá suporte a cerca de 130 tipos de arquivos *raster*, enquanto a OGR, a aproximadamente 70 tipos de arquivos vetoriais. Ambas foram utilizadas na implementação do módulo responsável pela leitura e escrita de arquivos do sistema desenvolvido. Os testes realizados usaram os formatos de arquivos raster

TIFF/GeoTIFF (.tif), GRASS ASCII GRID, GRASS Rasters, JPEG (.jpg) e Erdas Imagine (.img) e o formato de arquivo vetorial Shapefile (.shp).

### 3.1.5 Sistema Gerenciador de Janelas Qt

O *framework* Qt é um do *toolkit* gráfico multiplataforma escrito em C++, muito usado na indústria de software de alto desempenho para a elaboração de interfaces gráficas orientadas ao usuário (*Graphical User Interfaces* - GUIs) e pode ser considerado um padrão da indústria, possuindo um grande conjunto de *widgets* (controles na terminologia *Windows*). Também é importante ressaltar que as tecnologias Qt, OpenGL e OSG estão perfeitamente integradas, não sendo necessário fazer, por exemplo, nenhum tipo de conversão para mostrar uma figura (nesse caso um terreno) renderizada pelo OpenGL/OSG numa janela do Qt, aumentando a eficiência e a robustez do sistema. O Qt foi utilizado na implementação da interface gráfica do sistema desenvolvido.

### 3.1.6 Programação Orientada a Objetos

O uso da orientação a objetos é particularmente conveniente para o desenvolvimento de aplicações gráficas de uma maneira geral. Isto porque, componentes desses programas podem ser implementados na forma de objetos intuitivos (O'DOCHERTY, 2005).

### 3.1.7 Programação Genérica

Em alguns casos pode não ser vantajoso, no desenvolvimento de software SIG, o acoplamento entre os algoritmos e estruturas de dados fornecido pelo desenvolvimento de software orientado a objetos, isto por que, um grande número de algoritmos não depende da implementação de uma estrutura de dados em particular, mas apenas de algumas propriedades semânticas fundamentais destas como a habilidade de passar de um elemento da estrutura ao próximo ou de comparar dois elementos (VINHAS *et al.*, 2002).

A programação genérica é voltada para a construção de algoritmos que independam de um determinado tipo ou estrutura de dados e que sejam tão eficientes (com relação à velocidade de execução e utilização de recursos) quanto, a implementação desses mesmos algoritmos acoplados a determinados tipos ou estruturas de dados.

### 3.1.8 Padrões de Projeto

Com o objetivo de se conseguir um melhor aproveitamento dos paradigmas de programação orientada a objetos e genérica, foram utilizados os seguintes padrões de projeto no desenvolvimento do sistema proposto:

*Padrão de Projeto Iterador*: utilizado no acesso ao conjunto de vértices, arestas e faces (elementos triangulares) contidos na estrutura de dados responsável pela representação computacional da TIN e também no módulo responsável pela leitura de arquivos *raster* e vetorial.

*Padrão de Projeto Factory*: utilizado no desenvolvimento do módulo responsável pela leitura de diferentes formatos de arquivos *raster* e vetorial;

Além dos padrões de projeto citados acima, também foram necessários os seguintes conceitos durante o desenvolvimento do sistema:

*Handler*: utilizado no acesso individual aos elementos vértices, arestas e faces armazenados na estrutura de dados utilizada na representação computacional da TIN;

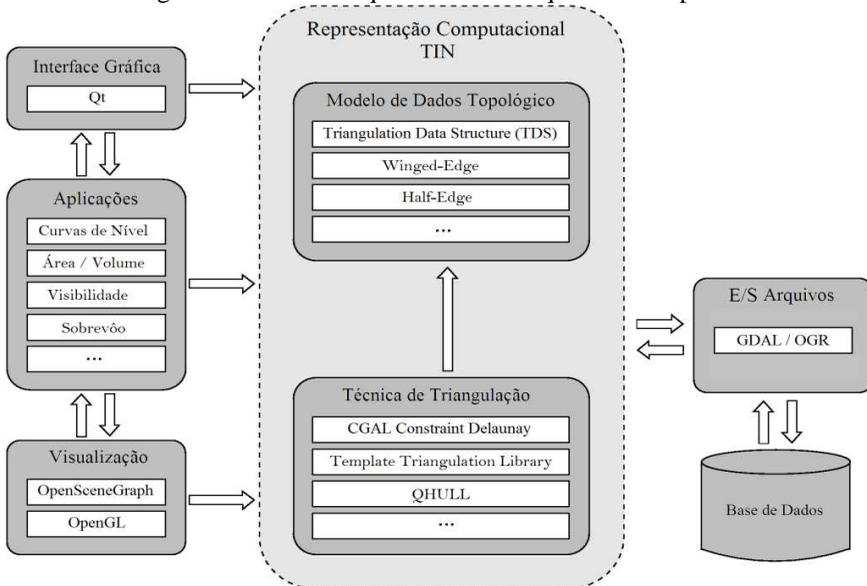
*Circuladores (Circulators)*: utilizado na implementação das classes responsáveis pelo acesso às relações de conectividade dos vértices e faces da triangulação;

*Máquina de Estado Hierárquica (Statechart)*: todo o controle da interface gráfica do sistema desenvolvido foi criado utilizando máquinas de estado.

### 3.2 Arquitetura do Sistema

A Figura 5 ilustra a representação diagramática dos componentes da arquitetura proposta, assim como a integração entre eles. Como pode ser observado, foi feita uma divisão conceitual do sistema em cinco módulos: Representação Computacional TIN, E/S Arquivos, Visualização, Interface Gráfica e Aplicações.

Figura 5 - Modelo Esquemático da Arquitetura Proposta.



#### 3.2.1 Módulo de Representação Computacional da TIN

Responsável pela criação de uma TIN por meio da técnica de Triangulação de Delaunay e pela sua representação computacional por meio de uma Estrutura de

Dados Topológica (EDT). É interessante observar que todos os outros componentes do sistema têm acesso a esse módulo sem a necessidade de se preocupar como a TIN é construída ou representada, ou seja, independentemente da biblioteca de triangulação ou EDT utilizada esse módulo oferece funcionalidades de como gerar um modelo TIN; fornecer uma representação explícita para as entidades vértices, arestas e faces do modelo; permitir consultas eficientes relacionadas às relações de conectividade existentes na topologia da triangulação gerada. Além disso, possui um alto grau de customização, na medida em que permite ao usuário adicionar informações às entidades vértice, aresta e face do modelo TIN por meio de programação genérica.

O módulo “Técnica de Triangulação” é o responsável por gerar de fato a TIN utilizando-se da técnica de Triangulação de Delaunay com *Constraints* (linhas de restrição) fornecida pela biblioteca CGAL. O pacote de triangulação da CGAL também oferece funcionalidades como localização de pontos na subdivisão planar induzida pela malha triangular e inserção e remoção de vértices na triangulação. Futuramente, esse módulo poderá ser construído a partir de outras bibliotecas de software de geração de malhas triangulares como, por exemplo, a TTL (Template Triangulation Libray) e a QHULL, considerando que ele foi projetado para funcionar como uma camada de acesso às funcionalidades de implementações de bibliotecas de triangulação, escondendo toda a sua complexidade associada e fornecendo uma interface amigável ao usuário.

O módulo “Modelo de Dados Topológico” se responsabiliza pela representação computacional de toda a topologia relacionada à triangulação utilizando a estrutura de dados topológica TDS (Triangulation Data Structure) fornecida pela biblioteca CGAL. A TDS oferece funcionalidades relacionadas à topologia da triangulação como, por exemplo, encontrar todas as faces incidentes em um dado vértice ou todas as faces adjacentes a uma face. Futuramente, esse módulo também poderá ser construído a partir de outras estruturas de dados topológicas como a *winged-edge modificada*, *quad-edge* ou a *half-edge*, funcionando como uma camada de acesso às funcionalidades de diferentes implementações de estruturas de dados topológica.

### 3.2.2 Módulo E/S Arquivos

Uma questão fundamental na proposta de elaboração de um novo sistema de software de modelagem digital de terrenos é a sua compatibilidade com aplicações já existentes. Logo, o sistema deve ser capaz de suportar a entrada de dados, tanto no formato vetorial (arquivos de pontos irregularmente espaçados ou curvas de nível), como no formato *raster* (matrizes de elevação), garantindo assim a sua integração com outras plataformas de software.

O módulo “E/S Arquivos” engloba todas as funcionalidades de leitura e escrita (E/S) de diferentes formatos de arquivos *raster* e vetoriais no sistema desenvolvido, garantindo assim a sua integração com outros sistemas computacionais. Como é mostrado na Figura 5, esse módulo foi desenvolvido utilizando as bibliotecas de

software GDAL e OGR funcionando como uma camada de acesso única a elas e escondendo grande parte de suas complexidades.

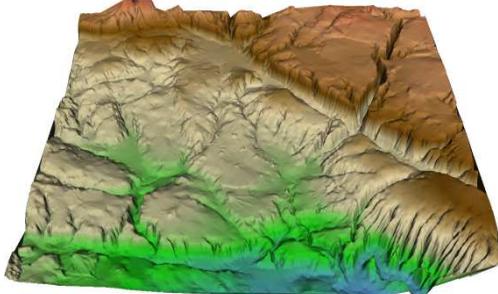
### 3.2.3 Módulo Visualização

É o responsável pela visualização em perspectiva tridimensional de MDTs, representados a partir de TINs geradas pelo sistema e que fazem uso intensivo das bibliotecas OpenGL e OSG. Foram implementados os modos de renderização por pontos, *wireframe* e sólido. Na renderização por pontos, são mostrados apenas os vértices da TIN. Na renderização *wireframe* são exibidas apenas as arestas da TIN e, finalmente, na renderização sólida são desenhadas as faces triangulares com a aplicação de modelos de iluminação (*Flat Shadding* e *Gouraud* disponibilizados pelo OpenGL) para dar uma melhor noção de profundidade na imagem gerada. Também é utilizada na renderização a interpolação de cores com o intuito de melhorar a compreensão da variação de altitude ao longo do terreno (vide Figura 1b).

Como a TIN gerada para representar um MDT pode ser bastante complexa, podendo chegar à ordem de um milhão de faces triangulares, é necessário o uso de técnicas eficientes de renderização. No sistema, todos os dados de renderização referente a TIN (vértices, arestas e faces) são guardados diretamente no hardware gráfico (nesse caso a placa de vídeo). Desta maneira, toda vez que o OpenGL necessitar de dados para a renderização, fará acesso direto a eles na memória do *hardware* gráfico, o que significa que os dados não precisarão ser transferidos da aplicação para a placa gráfica passando pela CPU, aumentando substancialmente a velocidade/eficiência da renderização da imagem.

Na Figura 6 é mostrada a imagem gerada de uma TIN, em perspectiva, com um total de 3.135.364 faces, sendo que uma delas foi guardada diretamente na memória da placa de vídeo, nesse caso, a NVIDIA GeForce 8600M GT. A renderização foi obtida com interpolação de cores e o modelo de iluminação *Gouraud*, sendo conseguida uma taxa de renderização de 28 quadros por segundo.

Figura 6 – Renderização de uma TIN em perspectiva com interpolação de cores e modelo de iluminação.



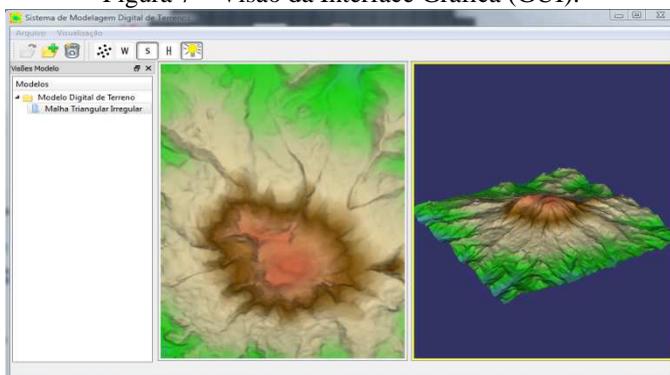
### 3.2.4 Módulo Interface Gráfica

Toda e qualquer tipo de interação do usuário com o sistema (abrir arquivos, executar a triangulação, etc.) foi realizada por meio de componentes GUI (menus, janelas, botões, etc.), integralmente gerenciados com o auxílio da biblioteca Qt.

O sistema desenvolvido também permite ao usuário executar uma navegação 3D no terreno modelado em tempo real por meio de um *trackball* virtual. O *trackball* virtual é um método de interface com o usuário que permite rotações arbitrárias de modelos tridimensionais utilizando cliques e movimentação do mouse em uma janela de visualização bidimensional.

Na Figura 7 é possível observar a janela principal do sistema desenvolvido, com a exibição de duas vistas de um MDT: uma de cima (esquerda) e outra em perspectiva (direita).

Figura 7 – Visão da Interface Gráfica (GUI).



Apesar de ainda não estar integrado no sistema desenvolvido, outra preocupação foi a de manter o mesmo reativo (respondendo a eventos do usuário) durante processamentos intensos, neste caso, durante a geração da TIN (execução da triangulação). Esse problema pode ser solucionado pelo uso de duas *threads*, isto é, uma *thread* para a execução da interface gráfica, e outra *thread* para a execução da triangulação. Usando essa abordagem, a interface gráfica se mantém responsiva (não trava ou pára de responder a eventos do usuário) independentemente de qualquer processamento que esteja sendo executado. Desta maneira, a fim de verificar a validade dessa solução também foi desenvolvido um teste em que a interface gráfica e o método responsável por gerar a TIN são executados em *threads* diferentes.

### 3.2.5 Módulo Aplicações

Esse módulo contém todas as possíveis aplicações que poderão ser futuramente implementadas e integradas ao sistema utilizando-se das funcionalidades fornecidas pelos módulos anteriormente descritos como, por

exemplo, geração de curvas de nível, cálculos de área, volume e testes de visibilidade. Por meio dos recursos disponíveis junto aos módulos Representação Computacional TIN, E/S Arquivos, Visualização e Interface Gráfica, a intenção é que eles possam ser utilizados futuramente como blocos de construção para o desenvolvimento de diferentes aplicações relacionadas à modelagem digital de terrenos.

#### 4. TRABALHOS FUTUROS

Em termos de perspectivas futuras, novas funcionalidades podem ser acrescentadas ao sistema desenvolvido neste trabalho, como por exemplo: refinamentos adaptativos da malha triangular pela inserção de pontos adicionais de maneira a eliminar triângulos muito longos ou quase degenerados melhorando dessa forma a regularidade local e global da malha triangular associada a superfície gerada. A utilização de outras fontes de informações como linhas de quebra, linhas estruturais e linhas de falésia também podem aprimorar reconstrução da superfície do terreno, aumentando sua acurácia.

A renderização da TIN pelo sistema pode sua performance melhorada pela execução de uma etapa de pré-processamento dos dados (ex: agrupar espacialmente os vértices dos triângulos na forma de *triangles strips*) antes de mandá-los para a GPU e também por operações de simplificação da TIN gerando vários níveis de resolução da mesma (LOD – Level Of Detail) durante a etapa de renderização com o intuito de melhorar a sua eficiência (velocidade de execução).

#### 5. CONCLUSÃO

O presente trabalho apresentou os resultados da especificação e implementação da arquitetura de uma plataforma de software voltada para a modelagem digital de terrenos. Nele foi mostrado o nível de maturidade de diferentes componentes de software *opensource* disponíveis no mercado (CGAL, GDAL/OGR, OpenGL, OpenSceneGraph e Qt), e como eles podem ser utilizados em conjunto para a produção de aplicações complexas como o sistema desenvolvido.

Dentre todas as funcionalidades desenvolvidas, merece destaque o fato do núcleo de representação de uma TIN estar fundamentado sobre uma estrutura de dados topológica (TDS fornecida pela CGAL) que permite de modo eficiente uma série de operações de consulta sobre relações de conectividade (adjacência e incidência) entre as entidades topológicas vértice, aresta e face de uma TIN em tempo constante ( $O(1)$ ), isto é, proporcional ao número de entidades envolvidas, independente da sua dimensão (pontos amostrados sobre a superfície do terreno).

Apesar do projeto estar em uma fase inicial, os resultados alcançados até presente momento com a especificação e implementação da arquitetura de um sistema de modelagem digital de terrenos baseada em TIN utilizando e integrando diferentes tecnologias; foi verificada a viabilidade do seu desenvolvimento e de futuras aplicações.

**AGRADECIMENTOS:**

Os autores agradecem ao CNPq/CAPES pelo apoio financeiro à pesquisa com a concessão de bolsa para a realização deste trabalho.

**REFERÊNCIAS BIBLIOGRÁFICAS**

- ABDUL-RAHMAN, A. Digital terrain model data structures. *Buletin Ukur*, v. 5, p. 61-72, 1994.
- BARBOSA, R. L. *Geração de modelo digital do terreno por aproximações sucessivas utilizando câmaras digitais de pequeno formato*. Dissertação de Mestrado em Ciências - Faculdade de Ciências e Tecnologia/UNESP - Campus de Presidente Prudente, Presidente Prudente, dez. 1999.
- BERG, M. et al. *Computational geometry: algorithms and applications*. 3. ed. New York: Springer-Verlag, 2008. 386 p.
- DANOVARO, E. et al. Out-of-core multiresolution terrain modeling. In: BELUSSI, A. et al. (Ed.). *Spatial Data on the Web*. Berlin: Springer-Verlag, 2007. cap. 3, p. 43-63.
- EL-SHEIMY, N.; VALEO, C.; HABIB, A. *Digital terrain modeling: acquisition, manipulation, and applications*. Boston: Artech House, 2005. 257 p.
- HJELLE, O.; DAEHLEN, M. *Triangulations and applications*. New York: Springer-Verlag, 2006. 234 p.
- JONES, C. B.; KIDNER, D. B.; WARE, J. M. The implicit triangulated irregular network and multiscale spatial databases. *The Computer Journal*, v. 37, p. 43-57, 1994.
- KETTNER, L.; NAHER, S. Two computational geometry libraries: LEDA and CGAL. In: GOODMAN, J. E.; O'ROURKE, J. (Ed.). *Handbook of Discrete and Computational Geometry*. Second. Boca Raton, FL: CRC Press LLC, 2004. cap. 64, p. 1435-1463.
- LI, Z.; ZHU, Q.; GOLD, C. *Digital terrain modeling: principles and methodology*. London: CRC PRESS, 2005. 323 p.
- O'DOCHERTY, M. *Object oriented analysis and design: understanding system development with UML 2.0*. England: John Wiley & Sons Ltd, 2005. 559 p.
- PETRIE, G.; KENNIE, T. J. M. Terrain modelling in surveying and civil engineering. *Computer-Aided Design*, Newton, MA, USA, v. 19, n. 4, p. 171-187, 1987. ISSN 0010-4485.
- PITERI, M. A. et al. Triangulação de Delaunay e o princípio de inserção randomizado. II Simpósio Brasileiro de Geomática - V Colóquio Brasileiro de Ciências Geodésicas, Presidente Prudente - SP, p. 9, jul 2007.
- SULEBAK, J. R. Applications of digital elevation models. *DYNAMAP Project*, p. 11, 2000.
- VINHAS, L. et al. Programação genérica aplicada a algoritmos geográficos. In: IV Simpósio Brasileiro de GeoInformática, *Anais do IV Simpósio Brasileiro de GeoInformática*, Caxambu, MG, Brazil 2002. v. 1, p. 117-122.
- (Recebido em outubro de 2013. Aceito em Novembro de 2013).