OPEN SOURCE SYSTEM INTEGRATION FOR DIGITAL TERRAIN MODELING

F. F. Oliveira\textsuperscript{a}, M. A. Piteri\textsuperscript{b}, M. Meneguette Jr\textsuperscript{b}

\textsuperscript{a} Posgraduate Program in Cartographic Sciences, UNESP – Universidade Estadual Paulista, FCT - Faculdade de Ciências e Tecnologia Roberto Simonsen Street, 305, 19060-900 Presidente Prudente – SP, Brazil ffoplay@gmail.com

\textsuperscript{b} Dept. of Mathematic, Statistic and Computation, UNESP-Universidade Estadual Paulista – FCT Faculdade de Ciências e Tecnologia, Roberto Simonsen Street, 305, 19060-900 Presidente Prudente – SP, Brazil (piteri, messias)@fct.unesp.br

KEY WORDS: Opensource Integration System, DEM/DTM, Topological Data Structures, Algorithms, Visualization

ABSTRACT:

The representation of land surface has been a recurrent challenge for both scientific community and for the industry. Although we represent topographic surfaces with a good degree of accuracy by means of maps, they are not always the best tools to apprehend the whole complexity of some reliefs. More recently, with the gradual reduction of costs and development of computational techniques, the computer has allowed advances in the terrain representation. The commercial software systems that have features associated with the digital terrain modeling focus on specific problems, not addressing more general issues associated with various representation and computational manipulation of the terrain surface. Another important aspect is the lack of freeware products (opensource) in this area. In this sense, the greatest contribution of this work is the specification of the architecture for an open source system focused on the representation of digital terrain models based on TIN (Triangular Irregular-Network) that integrates different technologies, which can therefore support a wide range of applications. The implementation of the system follows the philosophies of the object-oriented programming and generic programming, two current paradigms that enable the integration of technologies such as CGAL, TTL, GDAL, OGR, OpenGL, OpenSceneGraph and Qt, which have become standard in software development, most of them available in public domain. Moreover, the representation core of the system works with multiple topological data structures, which can access in constant time, all nine adjacencies relationships between vertex, edge and face, in a planar triangular subdivision, no matter the size of the problem (mesh size). This is crucial since all application about terrain representation necessarily makes use of these connectivity operations.

1. INTRODUCTION

Getting to represent the land surface or part of it in a convenient way and with certain degree of accuracy, has been a challenge for the scientific community. The maps have been the most used media for this purpose. Modern maps are designed with a high level of mathematical rigor, allowing for spatial information (position, measures of distance) being extracted from them with good security (Li, 2005). They also possess a generally well defined as much as intuitively symbolic system, allowing the user, depending on the map type, to view specific information quickly at the same time that has a global scene, due to their high degree of generality.

The representation of the terrain surface in conventional mapping can use points, hypsometric color, shading, contour, among others. In particular, contour maps are the most usual static and two dimensional representation of a topographic surface. In this type of map, the land is represented by isolines of constant elevation in a specified interval. A major drawback in this form of representation is that the terrain values are represented only along isolines (anomalies of the land between the lines are not identified), requiring the use of interpolation methods to infer values of elevation between the contours (El-Sheimy, 2005).

There is also a loss of information due to the fact the maps use a two dimensional representation for a three dimensional reality.

Moreover, as we can see by Figure 1a, the withdrawal of three-dimensional information (contained only implicitly in this kind of maps) is not intuitive, so that the user has to build up a mental imagery of the 3D reality (Figure 1b). From this limitation emerges the need for a 3D representation of the earth’s surface.

![Figure 1](image_url)

Figure 1 - Terrain representations: (a) 2D; (b) 3D.

With the intensive use of the computer in cartography and the consequent facilities for storage, processing and visualization of the mapping data, it became possible to develop and create digital models suitable for the representation and computational manipulation of the land surface in a way that conveys more easily the spatial characteristics to the users.
The existence of a Digital Terrain Model (DTM) associated with a portion of the land surface allows a number of operations to be carried out on it, as for example: extraction of contour lines and drainage networks, projects of building roads, tunnels and dimensioning of soil displacement; and also environmental impact studies, such as the computation of flooded areas in the construction of hydroelectric dams and the volume of the reservoir. Another very interesting application is associated with games and flight simulation that is a fundamental activity in the training of pilots.

As can be seen, the production and use of digital terrain models are of fundamental importance in planning activities, aid to decision making and the development of projects related to areas such as GIS (Geographic Information Systems), Engineering Mapping, Environmental, Military, Civil (El-Sheimy, 2005; Li, 2005).

That said, the central objective of this study is to describe the proposal for a software platform dedicated to the manipulation of the digital terrain models based on TIN, which integrates several technologies and, therefore, serve as the basis for the development of applications in different areas of knowledge.

2. DIGITAL TERRAIN MODELING

From a general perspective the problem of Digital Terrain Modeling can be divided into three main areas: Acquisition, Manipulation and Applications.

Acquisition includes the collect and digitalization of a set of representative samples of the ground. The data for drawing up the models can be obtained from maps, photogrammetric techniques, surveying, by means of radar systems, laser, etc. The step of acquiring the data form the basis of all operations that may be made on the DTM, and the greater the accuracy of the collected data, the best shall be the model.

In order the data can be used in a convenient way generating the necessary information for a particular application, it must be computationally organized (data structure) to allow efficient retrieving and extraction of relations. The stage of manipulation comes mainly from construction and representation of the computed terrain, that is by the topological and combinatorial structure built from the sampling done in the previous phase. The computational data model used by a DTM should allow basic operations on it, such: modifications, refinements and procedures for converting between different data structures that can be used to represent them (El-Sheimy, 2005).

The applications are procedures of analysis executed on the DTM as: visibility analysis between points, volume and area calculations, generation of slope maps, viewing the model in perspective, profiles and cross sections, among others. For now the focus of this paper will concentrate specifically on the manipulation.

2.1 Data Models

Among the many existing methods that can be used to obtain a representation of topographic surfaces in a digital format, the most used are: Regular Grid and Triangular Irregular Network (El-Sheimy, 2005; Li, 2005).

In the Regular Grid data model, the terrain surface is represented by a matrix data structure where each entry contains an elevation value \( z \) sampled at regular intervals in \( x \) and \( y \) plane. This is a model for easy storage and computer manipulation, since all topological relations between points have been implicitly saved in the data structure. This model also has an easy integration with raster data and accepts algorithms for processing and analysis of surface. However it can have a large number of redundant data in flat areas, and can not accurately represent areas with more complex relief (Figure 2a).

On the other hand, in the TIN data model the land surface is represented by a set of triangular polygons, such that the points forming their vertices are irregularly placed on the ground. This model can better represent the specific characteristics of more complex relief, since it can be used a larger amount of points in areas with more rugged terrain and a smaller amount of points in most flat areas (Figure 2b).

In addition to better represent the topography of a surface area, the TIN model can achieve a more efficient storage of data, as its size is proportional to the amount of data used to account for the DTM, what does not occur with the model grid, whose size is proportional to the considered area. The main disadvantage of the TIN model in relation to the Grid one is due to the greater complexity in the creation and management of the topological data structure used and the same for the procedures that work on the topology of the model aiming to extract the connectivity and adjacency relations among the triangular mesh elements.

![Figure 2 – Data Models based on: (a) Regular Grid; (b) TIN.](image)

3. PROPOSED SYSTEM

There are a relatively reasonable amount of commercial software systems that make use of, or contains tools for manipulation of DTM. Among them we mention: Topograph Idrisi, Envi, ArcGIS, LPS (Leica Photogrammetry Suite) and SOCET SET. However most of these systems are software oriented GIS with only modules for terrain modeling and therefore are not specific for this purpose. Even so, the DTM modules provided by those software, often focus on specific applications such as road modeling project or drainage models. That leads to the need of more than one software to accomplish a whole project related to digital terrain modeling. Another important factor is the lack of free software in this area. Despite the existence of consistent programs as Spring (Camara, 1996), Landsurf (Wood, 2008) or Topocal (Riquelme, 2004), they suffer the same problems related to use of commercial software, there is not one generic enough software to be used in a wide range of problems related to digital terrain modeling. Thus, the aim of this work is to propose the creation of a specific software system for digital terrain modeling.
3.1 Used Tools

The system is being developed using the programming language C++, initially on the Windows platform, using the compiler MicrosoftVisualStudio.

Considering the size and multidisciplinary nature of the project it shall be use free software multiplatform libraries, with different purposes, such as CGAL (Computational Geometry Algorithms Library), GDAL (Geospatial Data Abstraction Library), OGR (Simple Feature Library), OpenGL and OpenSceneGraph.

Each of these tools was chosen taking into account: performance, stability, robustness and integration in their area of expertise. Worth emphasizing that they are implemented in C/C++ and have become standard in the software development. In the next sections there is a more detailed description of the proposed system and the features of each of these libraries.

3.2 System Data Model

The topological hierarchical model to be used by the system for the representation of the triangular mesh, makes a separation between the components: geometry, topology and attributes. Thus, the values associated with the elevation of points sampled on a portion of the surface or the slope on a triangular face have geometric nature, while features associated to them, such as kind of soil, ownership, value of the property, land use and others, are associated with component of attributes.

On the other hand, the reconstruction of the surface will be made from the Delaunay triangulation on a set of irregular sampled points, leading to a combinatorial and topological structure (triangular mesh), which constitutes the core of the representation of the system (architecture) that is based on topological data structure. Indeed, the information related to the geometry and attributes are only components of the topology.

The use of a topological data structure for the representation of the triangular mesh has a prominent role in increasing the performance of the proposed system, because any information about connectivity or adjacency on its topology can be obtained in time constant or proportional to number of entities involved (regardless of the size of the problem), allowing to address the delays common to the Digital Terrain Modeling systems which are generally based on simple data structures, not suitable to manipulate large amounts of points.

Basically two abstract data (or interfaces) models are created representing the topological data structure that contains the triangulation and algorithm responsible for the creation of triangulation. These two models act as access interfaces (of the system data model) for implementations of topological data structures and algorithms of existing triangulation. For example, it does not matter to the system how the data model that represents the topology of the triangular mesh is implemented, if makes use of the modified winged-edge, half-edge (Chiyokura, 1988) or other topological data structure, it should only implement the interface of the model abstract data representing the topology of the triangulation, no matter how. This gives great flexibility to the system since, if we want to change the way the triangular mesh is computationally represented in future, for example, using topological quad-edge data structure (Guibas, 1985), we have only to re-implement the model associated with the topology, not need to change other parts of the system. Figure 3 brings a UML (Unified Modeling Language) class diagram that illustrates how you can separate the data structure representing the mesh, from the triangulation technique used which could be obtained by CGAL or TTL. In other words, how the abstract models of the topology data and triangulation are processed by the system.

![Figure 3 – Digital Terrain Model class diagram.](image)

Initially, the triangular mesh is performed with the aid triangulation package of CGAL, a library of software that offers to industry and academic community implementations of data structures and efficient and reliable geometric algorithms to solving basic problems in the area of computational geometry, such as incremental algorithms for Delaunay triangulation and Delaunay with constraints, thus facilitating the development of more complex geometrical applications (Kettner, 2004).

For the representation of triangulation it is used the topological data structure TDS (Triangulation Data Structure) also provided by CGAL. This structure is based on faces and was made exclusively for the representation of triangular meshes. Another option would be to use the library TTL (Triangulation Template Library). The option to use the CGAL is due to the fact it offers both the topological data structure to represent the triangulation as well as the algorithms to generate it. In the TTL case it is only provided tools to generate the triangulation, being left to the user to create a data structure that represents it (Hjelle, 2006).

3.3 Integration with Existing Systems

A key issue in a proposed new software system for digital terrain modeling is its compatibility with existing systems. Therefore, the system should be capable of supporting the entry of data, both in vector format (such as files of irregularly spaced points, or break lines) as well as a type of raster (matrix of elevation), thereby ensuring an integration with other software platforms. To be more precise, the support for various formats of GIS raster and vector spatial data will be done with the help
of libraries GDAL and OGR, respectively, which offer many facilities for conversion between these formats (Warme"rdam, 2008).

### 3.4 Visualization

One of the most important components in GIS systems is the visualization module, since over 80% of information obtained by humans occurs by means of their visual system (Li, 2005). In systems of digital terrain modeling, visualization of the surface being modeled from several projection techniques and points of view is also important and helps the understanding and evaluation.

In the system under development, the various visualization processes on the represented ground are done through the API (Application User Interface) OpenGL, which is an interactive graphic system, fast, portable and has many resources to display three-dimensional objects, permitting the user to write programs that directly access the graphics hardware (Shreiner, 2004). However, this library does not offer any kind of tool in terms of overall management of 3D scenes. Thus, it will use the OpenSceneGraph, which is a graphical toolkit that builds a graph of the scene on OpenGL, allowing for assembly and efficient management of 3D scenes (Martz, 2007). The use of this library facilitates the visualization development for the system because it has features such as tests of mouse intersection with the scene (user interaction with the 3D scene), text rendering in 3D scenes, billboarding and routines for images/textures loading. Furthermore, the OpenSceneGraph also has tools for creating primitives particularly efficient for the rendering of triangulations as triangles strips and triangle fan (Hjelle, 2006), and routines of visibility and elimination of vertices (Frustum Culling, Occlusion Culling, Level of detail) thus reducing the total number of vertices transferred to the video card significantly improving the performance and interactivity of the user with the system which are critical in any software system.

### 3.5 User Interface

The user interaction with the DTM as well as its graphical representation is fundamental for a good understanding of it (through visual analysis). In particular, operations such as drag-drop, click, zoom and pan have a prominent role in this visual exploration. All and any iteration of the user with the system developed is done through GUI components.

More precisely, all aspects of the creation and windows management are made with the aid of the development environment Qt, which can be regarded as an standard for high performance, with a large set of widgets (controls in the Windows terminology) that are widely used in nowadays applications, such as menus, context menus, buttons, check box, and dockable toolbars which are most used components in this project. It is also important to emphasize the fully integrated Qt and OpenGL technologies. So, no needed exists for any conversion when displaying a picture (in this case a terrain modeled) rendered by OpenGL in Qt window, further increasing efficiency and robustness of the system.

### 3.6 System Architecture

Separately, all the discussed technologies does not make any sense for the system, so we must create a logical structure to make these technologies "talk between them". Considering the size of the project and the diversity of aggregate components to its constitution, the system conforms to the paradigms of object orientation and generic programming. Object Oriented Programming (OOP), offers a powerful mechanism to deal with the computational complexity of a project, making programs clearer, safer and promoting the maintenance of it (O’Docherty, 2005). Generic programming provides several tools related to the reuse and integration of source code (Alexandrescu, 2001).

In order to maximize these programming paradigms, the system makes intensive use of "design patterns", such as Model-View-Controller, Signals and Slots, Template, Iterator and Visitor, among others. This decision is not only to explore the use of both paradigms in full, but also to facilitate the maintenance of the system, the components integration and also their future extensions. Figure 4 gives an overview of the system architecture and the components integration. As can be seen, the topological and combinatorial structure of the triangulation (TIN) that represents the terrain is built from the input data supplied by the module "I/O Services".

![Figure 4 – Schematic Model of the System Architecture.](image-url)
4. CONCLUSION

Considering the lack of software tools focused exclusively on digital terrain modeling, it is urgent the need for new computer systems that address this problem. The proposed system aims at not only provide a new option for opensource tool, but also as a platform for development of software capable of supporting a wide range of applications that make use of digital terrain models. The system is based on the principles of the object oriented programming and generic programming.

Issues of performance, efficiency and robustness of the system are being considered and treated by the use of topological data structures and advanced techniques of existing triangulation in CGAL library. The development of the interface and the display aspects of the terrain or part of it, use open source standard libraries but are also object of this research. An additional challenge for the implementation of the system has been the integration between the various libraries used, in particular those concerning the techniques of generation and representation of the triangulation associated with the set of points. To resolve this problem, a layer of abstraction, following the general philosophy of generic programming was created, to facilitate the maintenance of the system and its extensibility.

In summary, this is an ongoing project and constitutes the development of a software platform aimed at Digital Terrain Modeling, which allows for the continuous build up of new applications in different fields of knowledge.

In continuation of the project, emerging issues such as multi-scale representations, constraints Delaunay triangulations, contouring from DTM and others, shall be object of research.

5. REFERENCES


