

# INTERACTIVE MULTI-LAYER TERRAIN VISUALIZATION TOOL

**Rafael Moreira Savelli and Paula Frederick**

Computer Graphics Technology Group – TECGRAF  
Pontifical Catholic University of Rio de Janeiro - PUC-Rio,  
Rua Marquês de São Vicente 255, Rio de Janeiro, Brazil  
{savelli, Paula}@tecgraf.puc-rio.br

**Roberto de Beauclair Seixas**

Institute of Pure and Applied Mathematics - IMPA  
Estrada Dona Castorina 110, Rio de Janeiro, Brazil  
rbs@impa.br

## Resumo

Há tempos em que pesquisadores da área da computação gráfica de todo o mundo lidam com uma infinidade de problemas de visualização de terrenos. Entretanto, alguns desafios recente tem demandado o uso de terrenos complexos os quais precisam ser carregados e visualizados em aplicações específicas. Assim, este trabalho propõe uma aplicação para visualização de terrenos complexos e de múltiplas camadas que permite uma interação fácil e ágil com o usuário. Nós também mostraremos algumas aplicações práticas de como os Fuzileiros Navais estão tirando proveito dela como parte do treinamento e simulação adotados pela Marinha do Brasil.

**Palavras-Chaves:** Simulação e Treinamento; Terrenos de múltiplas camadas; Visualização de Terrenos.

## Abstract

There has been a long time where computer graphics researchers from all over the world deal with a variety of terrain visualization problems. However, some recent challenges often demand much more complex terrain information that must be handled by a good terrain viewer application. So, this work proposes a multi-layer terrain visualization application which can handle complex terrains data in a very user-friendly and interactive way. We will also show some of its applications and how Brazilian Marine is currently benefiting from it as part of simulation and training processes.

**Keywords:** Simulation and Training; Multi-layers Terrains; Terrain Visualization.

## 1. INTRODUCTION

One large branch of computer graphics researching is digital terrain modeling and constructing. For years, researchers from all over the world have been publishing a considered variety of such works trying to understand and solve many real world problems. Even when digital satellite images became popular, about ten years later, vectorial terrains keep their growth and continued to be very useful in many sort of applications.

Today, as hardwares become faster and affordable, these works also evolved to a next generation. This phenomenon can be verified by just noting that most of them are now dealing with larger terrain data, carrying huge amount of layers and having both two- as well as three-dimensional applications. So me of these recently works are [1–4].

In our case study, complex situations that involve military activities such as tactical planning and decision making was joined into this visualization tool helping Brazilian Marine to indoctrinate and train its troops through simulations. In addition, multi-platform and lightweight applications are prerequisites to maximize usage among all forces departments

specially those with low-cost computers due to tight annual budgets.

To meet and handle these military complex situations, we will present an interactive user-friendly tool designed for visualizing and manipulating vectorial terrains. In order to accomplish this task, we put together many available technologies which are presented on section 2. Next, on section 3 the *hillshade* algorithm is shown. The resulting project and its applications are fully described on section 4. At last, in conclusion, we share some advantages and drawbacks transposed during the process. We also discuss in this section some new possible future features.

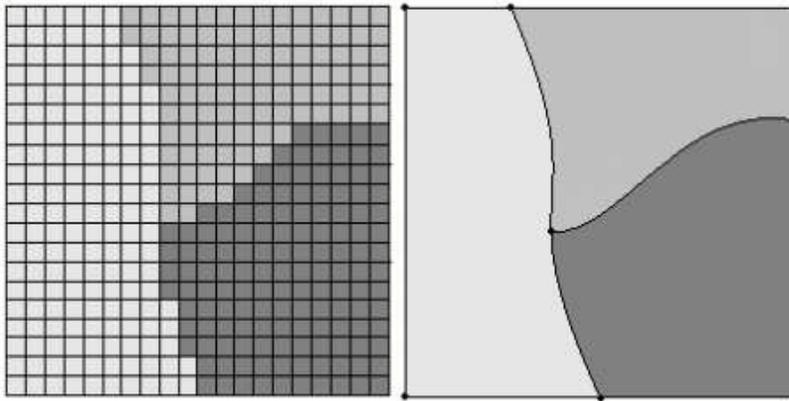
## 2. TECHNOLOGIES

To make this visualization tool possible, we need to combine a variety of functionalities, mainly by using several libraries. Every library has its own interface and performs internally one or more specific activities to the main tool. So, during this whole section, we describe all used libraries showing where to get them, how they work in our favor, comment possible successes and drawbacks as well as how we transposed them.

### 2.1. SHAPE

Due to limited memory capacity in every computer system, a real world terrain must be modeled and digitalized as being just a discrete form of its original physical terrain [5]. This discrete information is also known as *sample* and its frequency is determined depending on the issue. Another usual name for this type of representation is also known as *raster*. Large raster terrains can reach huge amount of samples, requiring large data disks to be stored in.

Another form of terrain representation involves primitives manipulation. Using both lines and arcs as well as basic polygons such as triangles and squares we associate attributes which best describes that terrain portion. Some of these attributes are: *height*, representing mountains and valleys. *Color*, blue for lagoons and rivers and green for some specific type of vegetation such as grass or broadleaved trees. These type of terrain is known as *vectorial* terrains. In figure 1, both raster and vectorial terrain representation are shown.



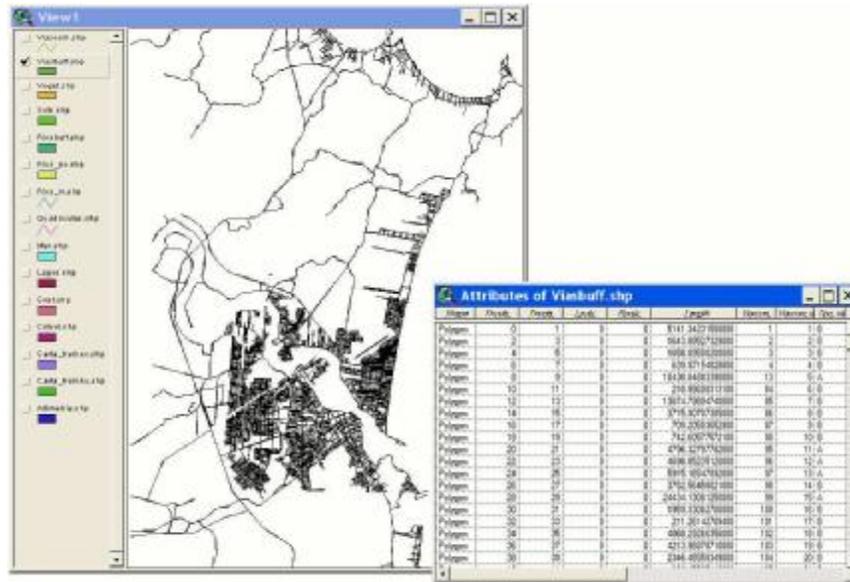
**Fig. 1.** Usual terrain representation: raster (*left*) and vectorial (*right*).

The preferred form of terrain representation was vectorial. The Brazilian Marine picked up this type of modeling for two main reasons: first, there are no quality losses even when zooming-in higher percentages rates. Second, they can store terrain data in *shapefiles* [6] and use *shapelib* [7] to access those files. Shapefiles give support to georeferential terrains as well as their numerous attributes. Shapelib however, provides an API to C++, which gives a way to

handle and manipulate these shapefiles in this powerful programming language.

## 2.2. TERRALIB

The TerraLib project [8] is an open-source GIS software library. Its main design goal is to support large-scale applications using socio-economic and environmental data. It is so powerful that supports coding of geographical applications using spatial databases, storing data in different DBMS including *MySQL* and *PostgreSQL*. Due to that storing policy, programmers can retrieve information by just querying on the database. Figure 2 shows how polygons dispose themselves in the form of tables. This library is developed in C++ and has programming interfaces also in java and visual basic [9].



**Fig. 2.** A typical Terralib project: More than four thousand polygons disposed in form of tables.

There are some noteworthy project based on TerraLib library. One of them is the TerraView [10], an application which provides functions for data conversion, display, exploratory spatial data analysis and spatial and non-spatial queries. Another great TerraLib usage can be noticed in the TerraAmazon project [11]. TerraAmazon is Brazil national database for monitoring deforestation in the Amazon rain forest, which manages more than 2 million complex polygons and 60 gigabytes of remote sensing images.

In our case study, we know that Brazilian Marine is always up dating territory information by altering their terralib data files. So, such visualization application should at least give support to those files allowing to import them in a fast and easy way. As expected, this library becomes mandatory in the project.

## 2.3. IMAGING, CANVAS DRAW AND USER INTERFACE LIBRARIES

In addition to those previously cited libraries, we also need to add some more to help us handle with images, canvas drawing and user interfaces. To reach each needs, we preferred to use tools provided from TeCGraf/PUC-Rio laboratory [12]. TecGraf is an important computer graphics research group that has been helping to solve many problems in the computer graphics field. Their projects are usually based and developed in conjunction with specific graphics libraries. Some of these libraries are: the Imaging Manipulation (or IM) [13], Canvas Draw (or CD) [14] and Portable User Interface (or IUP) [15].

The IM library is useful to manipulate many types of images, transparency when supported, I/O file stream, image capturing and many other hard tasks. The CD library gives support to viewport manipulation, coordinate system conversion as well as primitives drawing such as circles, lines, polygons and the like. At last, the IUP library makes possible to add user components such as buttons, drop-down lists, text fields and similar. Every component in this library has its own callbacks and attributes to expand their functionality and provide edibility.

Perhaps some of the most beautiful aspects about these all three libraries are the fact that they are lightweight, portable and free.

### 3. THE HILLSHADE ILLUMINATION ALGORITHM

The last feature presented is the hillshade illumination algorithm [16]. The Hillshade function obtains the hypothetical illumination of a surface by determining illumination values for each cell in a raster. It does this by setting a position for a hypothetical light source and calculating the illumination values of each cell in relation to neighboring cells. To keep it simple, the general hillshade illumination algorithm can be basically described in equation 1 as following:

$$\text{hillshade} = 255.0 * ((\cos(\text{zenith}) * \cos(\text{slop})) + (\sin(\text{zenith}) * \sin(\text{slop}) * \cos(\text{azimuth} - \text{aspect})) \quad (1)$$

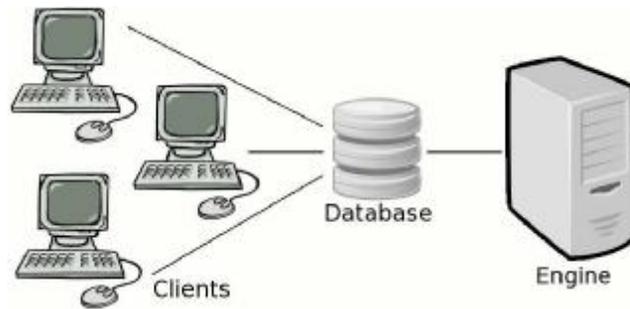
Where, *zenith*, *slop*, *azimuth* and *aspect* are all four radians angles. The *altitude* radians angle is the light source angle formed with the ground while *zenith* angle is the complement to that *altitude* angle. The *azimuth* angle is defined as being the direction of the illumination source. The *slope* is defined as being the steepest downhill descent from each cell in the surface. Finally, the direction the steepest downslope direction is facing is the *aspect*. The *Aspect* is then defined in the range of 0 to 2pi, with 0 toward east.

The hillshade algorithm effects are usually interesting because it can greatly enhance the visualization of a surface as seen in Figure 7. This kind of effect is very useful for Marines because the terrain illumination varies from time to time. As expected, shadows at seven hours in the morning are not the same ones from five afternoons. This is a nice feature that gave us excellent feedback from many users.

### 4. THE VISUALIZATION TOOL – VIS2D

As part of all Brazilian Marine training process, they must get familiarized with simulators and games. One of these simulators is the *Didactic Game System* or SJD [2]. In this simulator, each players gain a mission, a certain quantity of troops, radios, guns and ammunition. Then, they must traverse from one terrain point to another, respecting and attending all mission requisites. If an enemy appear in the pathway, is up to them to decide whether fight, back up or change their initial course. If obstacle is a large river for motorized troops, the process also involves similar decisions. These kinds of decisions making are some of the most important aspects in the Marine training process. This entire procedure including receiving a mission, planning and acting accurately despite setbacks is called military operation.

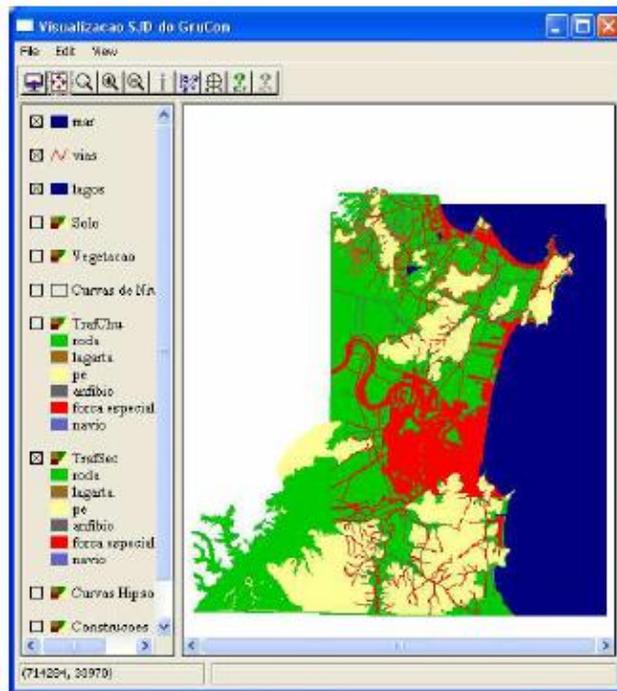
The employed system architecture adopted in this game can be viewed in Figure 3. The entire system works in cycles, so when engine needs to process all clients requests, it first reads those tasks from a database previously linked to that system. Secondly, it executes those tasks and after all work done, the outputs must be placed back again in the database to be interpreted by clients at their respective running cycles.



**Fig. 3.** The *Didactic Game System* architecture.

As expected, the database keeps all necessary information during the game. So, in every time period, the visualization tool *Vis2D* accesses these information by just reading directly from the database. Through ODBC configuration, it is possible not only to make *Vis2D* loads a multi-layered terrain from shapefiles but also shows current combat elements positions, ammunition left, radio battery life and the like. This kind of combined visualization turns this application in a very powerful tool to Brazilian Marine. It permits in a single user interface to keep an eye in every operation aspect during any simulation time.

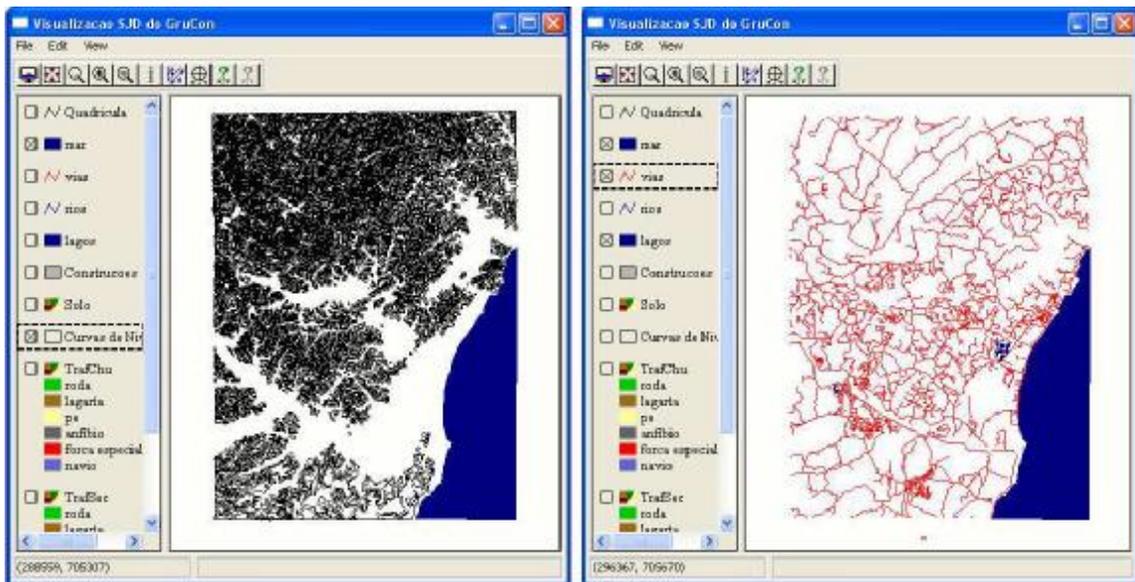
On the following Figure 4, we can observe how multi-layered terrain are manipulated and visualized in the *Vis2D* system. This figure reveals in the main canvas a trafficability terrain layer. By using colors, the trafficability layer evidences where a certain combat element can transpose or not from one region to another. Among other functionalities, it is very useful to avoid combat vehicles to get stuck in the mud.



**Fig. 4.** *Vis2D* overview system.

The left part of this interface keeps a layer list for every terrain attribute in which can be turned on or off. By clicking on the black boxes attached to each attribute, user can activate or deactivate sea viewing, urban areas, lakes, paved roads, and the like. To illustrate such behavior, Figure 5 shows just a couple of these layers for both paved roads and level curves. They were applied to a certain Brazilian east region where these attributes are abundant. If one drags a terrain layer and drop it few layers above or below, this layer gains or loses priority in

canvas drawing. This is useful to grant user the ability to paint for last what is most relevant or important to him.



**Fig.5.** Some Vis2D layers. *Left: terrain level curve and right: Paved roads.*

Next, on figure 6, we can view the *Vis2D* application running with some military information shown directly from its interface. Currently simulation status such as location, date time, terrain georeferential information as well as combat elements are all dynamic, taken from database every period of a time.

## 5. CONCLUSIONS

This multi-layered terrain viewer solution proves to be a very economical and efficient way to provide high quality training, cutting expenses to the extent that food, accommodation and transportation are no longer necessary when performing these very time consuming activities. The Brazilian Marine is now extensively using this application as part of forming new officials. They could run this application both in windows as well as linux platforms, gaining a substantial group of users, from many different forces departments.

No doubt that the most difficult part encountered during *Vis2D* development was studying and placing all libraries to work together, taking the most from each of them. This gave an odd improvement to the tool, but sometimes libraries conflicts were very hard to solve.

As shapefiles are always being improved by Brazilian Marine, maintaining and supporting the *Vis2D* application turns to be an indispensable task. Every time a new attribute is created in the shape database, *Vis2D* has to be adapted accordingly. This enforces both reliable and confident relationship between academic and Brazilian Marine force. This kind of healthy partnership has been keeping for years.

Lastly, we would like to thanks terralib project. This project main site is full of manuals, user guides as well as papers and posters in the GIS field. Once implemented, this library prove to be very efficient and functional, going beyond our initial expectations.

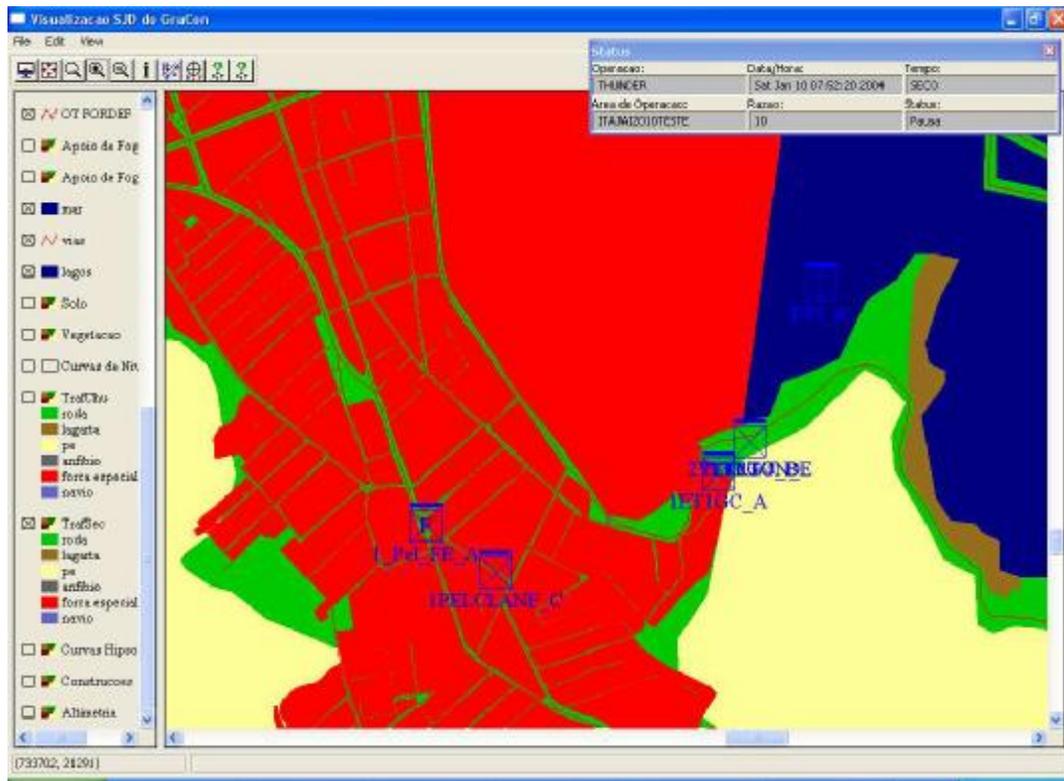


Fig. 6. Combat elements and status shown in Vis2D system.

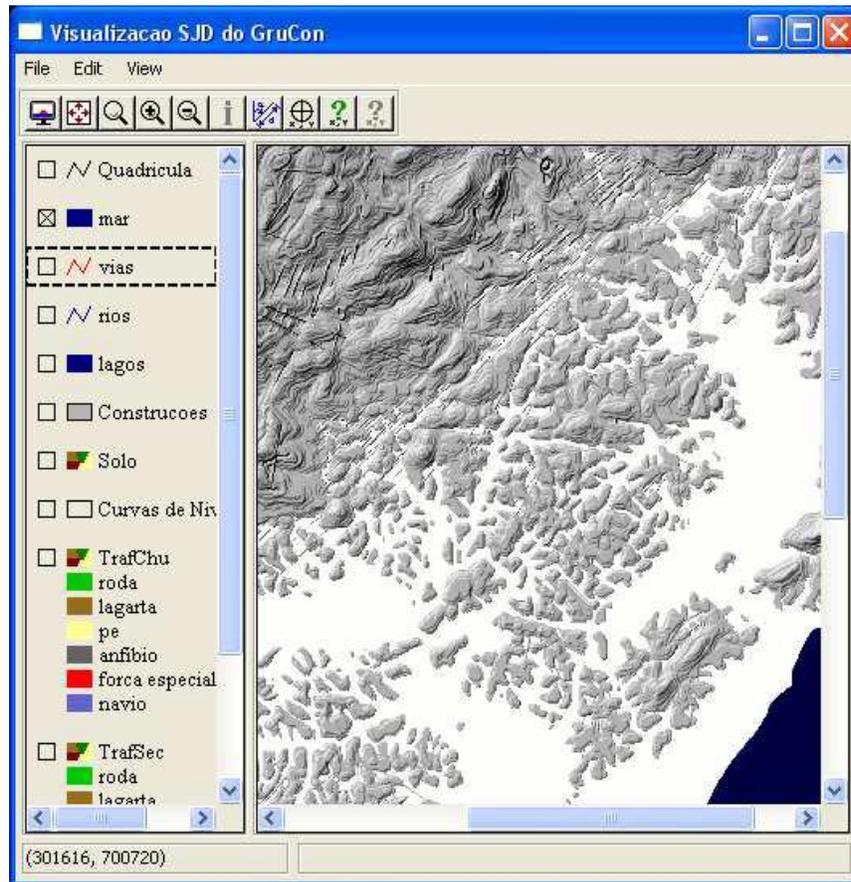


Fig. 7. Terrain hillshade for an approximate mid-afternoon time.

## 6. REFERENCES

- [1] Campos, D.V.; Seixas, R.B.: Command and Control: A low cost framework to remotely monitor military training. Proceedings of Spring Simulation Multiconference - SpringSim'11, in Military Modeling and Simulation Symposium - MMS, Boston, USA, 2011.
- [2] Lyrio, G.H.S.O; Seixas, R.B.: Didactic Games System: Fifteen years of development in military simulation. Proceedings of The North American Simulation and AI in Games Conference - GAMEON-NA, EUROSIS, Troy, USA, 2011.
- [3] Savelli, R.M.; Seixas, R.B .; Montenegro, A.A.: Objects Visualization in Digital Terrains Using Adaptive View-Dependent Techniques. Proceedings of the International Graphics on Computer Graphics Theory and Applications - GRAPP, Lisboa, Portugal, 2009.
- [4] Poyart, E.; Frederick, P.; Seixas, R.B.; Gattass, M.: Simple Real-Time Flight Over Arbitrary Sized Terrains. Technical Report 02/2002.
- [5] Worboys, M. F. and Duckham M.: GIS: A Computing Perspective 2nd edition, CRC P ress, Florida, 2004.
- [6] ESRI Shapefile Technical Description White Paper, Environmental Systems Research Institute, Inc.: Redlands, CA, 1998
- [7] Free C library for reading/writing shapefiles, V. 1.2, [shapelib.maptools.org](http://shapelib.maptools.org)
- [8] TerraLib Library Project, [terralib.org](http://terralib.org)
- [9] Câmara, G., Souza, R., Pedrosa, B., Vinhas, L., Monteiro, A., Paiva, J., Carvalho, M., Gattass, M.: TerraLib: Technology in Support of GIS Innovation. II Brazilian Symposium on GeoInformatics, São Paulo, 2000.
- [10] TerraView, [dpi.inpe.br/terraview](http://dpi.inpe.br/terraview)
- [11] TerraAmazon, [www3.funccate.org.br/geo//available/wiki-v01-TerraAmazon](http://www3.funccate.org.br/geo//available/wiki-v01-TerraAmazon)
- [12] Computer Graphics Technology Group, TeCGraf/PUC-Rio, [tecgraf.puc-rio.br](http://tecgraf.puc-rio.br)
- [13] Imaging Manipulation, [tecgraf.puc-rio.br/im](http://tecgraf.puc-rio.br/im)
- [14] Canvas Draw, [tecgraf.puc-rio.br/cd](http://tecgraf.puc-rio.br/cd)
- [15] Portable User Interface, [tecgraf.puc-rio.br/iup](http://tecgraf.puc-rio.br/iup)
- [16] Burrough, P. A. and McDonell, R.A., 1998. Principles of Geographical Information Systems Oxford University Press, New York, p. 190.