

VARIANT - Processing and Visualizing Terrains at Variable Resolution

Leila De Floriani, Paola Magillo

Dipartimento di Informatica e Scienze dell'Informazione – Università di Genova
Via Dodecaneso, 35 – 16146 Genova (Italy) - Email: {deflo,magillo}@disi.unige.it

Enrico Puppo

Istituto per la Matematica Applicata – Consiglio Nazionale delle Ricerche
Via De Marini, 6 – 16149 Genova (Italy) – Email: puppo@ima.ge.cnr.it

Abstract

We describe VARIANT (VARIABLE Resolution Interactive ANALYSIS of Terrains), a prototype system for processing and visualizing terrains at variable resolution, which is based on a multiresolution terrain model, called a Multi-Triangulation (MT).

1 Introduction

Terrain models used in geographic information systems are built from elevation values measured at a set of data points. A terrain model consists of a grid, having its vertices at data points: a predefined convention is used for interpolating the terrain surface within each grid element. Usually, grids composed of square or triangular elements are used. The accuracy of a terrain model increases with the resolution (density) of the grid, but this also increases the cost for manipulating, analyzing, and visualizing the model.

Different application tasks have different requirements in terms of accuracy of representation and/or efficiency of operations, and these requirements may change with time even within the same task: for example, in a flight simulator the accuracy needed in each part of the terrain depends on its distance from the moving viewpoint.

Multiresolution terrain models provide compact data structures, from which representations of the terrain at variable resolution can be retrieved on-line, according to the needs of each application task [1, 2, 8, 10, 12]. In [10], we proposed a general multiresolution model, called a *Multi-Triangulation* (MT), and in [3] we presented the main principles of its application to terrain modeling.

In this paper, we describe *VARIANT* (*VARIABLE Resolution Interactive ANALYSIS of Terrains*), a prototype system for analysis and visualization of terrains, based on the MT. *VARIANT* has been developed as a joint work of the Department of Computer and Information Sciences of the University of Genova, and of the Institute for Applied Mathematics of the National Research Council.

The system has a modular structure: the *MT-manager* encapsulates the data structure for the Multi-Triangulation, and provides operations for manipulating and inquiring it to the *MT-clients*. A special MT-client is the *MT-builder*, which builds the multi-triangulation off-line through a set of update operations provided by the MT-manager. Other MT-clients can use query operations provided by the MT-manager to perform computations such as terrain visualization, analysis (e.g., viewshed, drainage networks, path computations), simulations, etc.

Permission to make digital/hard copies of all or part of this material for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copyright is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires specific permission and/or fee.

GIS 97 Las Vegas Nevada USA

Copyright 1997 ACM 1-58113-017-1/97/11...\$3.50

An extended version of this paper [5] is available from http://www.disi.unige.it/research/Geometric_modeling/.

2 The Multi-Triangulation

Triangulated Irregular Networks (TINs), unlike regular square grids, do not rely on a regular distribution of the data points. This means that the resolution of a TIN can be locally adapted, either to the roughness of the surface, or to an error threshold variable over the domain. Thus, TINs are especially suitable to meet variable resolution requirements of application tasks: for this reason, they are at the basis of our multiresolution terrain model.

A *Multi-Triangulation* (MT) is made of a collection of *components* arranged into a partial order, which can be described by a *directed acyclic graph* (DAG) having such components as nodes. Each component holds two TINs, giving two different descriptions of the terrain over the same portion of the domain, at a lower and at a higher resolution, respectively. For convenience, we denote the TIN at lower and higher resolution of a node N with $low(N)$ and $high(N)$, respectively. An arc (N_a, N_b) is in the DAG if and only if one or more triangles of $high(N_a)$ also belong to $low(N_b)$, i.e., if $high(N_a) \cap low(N_b)$ contains at least one triangle. The DAG is rooted at a *source* component, holding only one TIN (the *high* one), representing the whole domain at lowest resolution available in the model. A further *drain* component is added to the DAG, which also holds only one TIN (the *low* one), representing the whole domain at the highest available resolution. An example of a multi-triangulation is shown in Figure 1.

Each triangle t appears in exactly two TINs, namely $high(N_a)$ and $low(N_b)$, in two components N_a and N_b connected by an arc (N_a, N_b) ; we say that triangle t is *associated with* arc (N_a, N_b) . Every triangle of an MT is associated with exactly one arc. A *cut* of the DAG is a set of arcs that separate the source from the drain. The set of triangles associated with the arcs of any possible cut in the DAG provide a TIN covering the whole domain: the closer the cut to the drain, the higher the resolution of the corresponding TIN. Having a cut either immediately above or immediately below a node N means using either triangles of $low(N)$ or triangles of $high(N)$ in the extracted TIN, i.e., such a TIN is made either less or more refined over the portion of domain corresponding to N . Therefore, the MT provides local control of the resolution, hence the possibility of extracting representations that have variable resolution over different zones of the domain. An example of a cut, and of its corresponding TIN, is shown in Figure 1. Formal definitions and properties of the MT can be found in [10, 3].

Every triangle in an MT is characterized by an approximation error, corresponding to the maximum vertical distance between the surface patch represented by the triangle and the original terrain. The accuracy of a TIN extracted from an MT depends on the error of its triangles, and, in general, it can be variable over the domain.

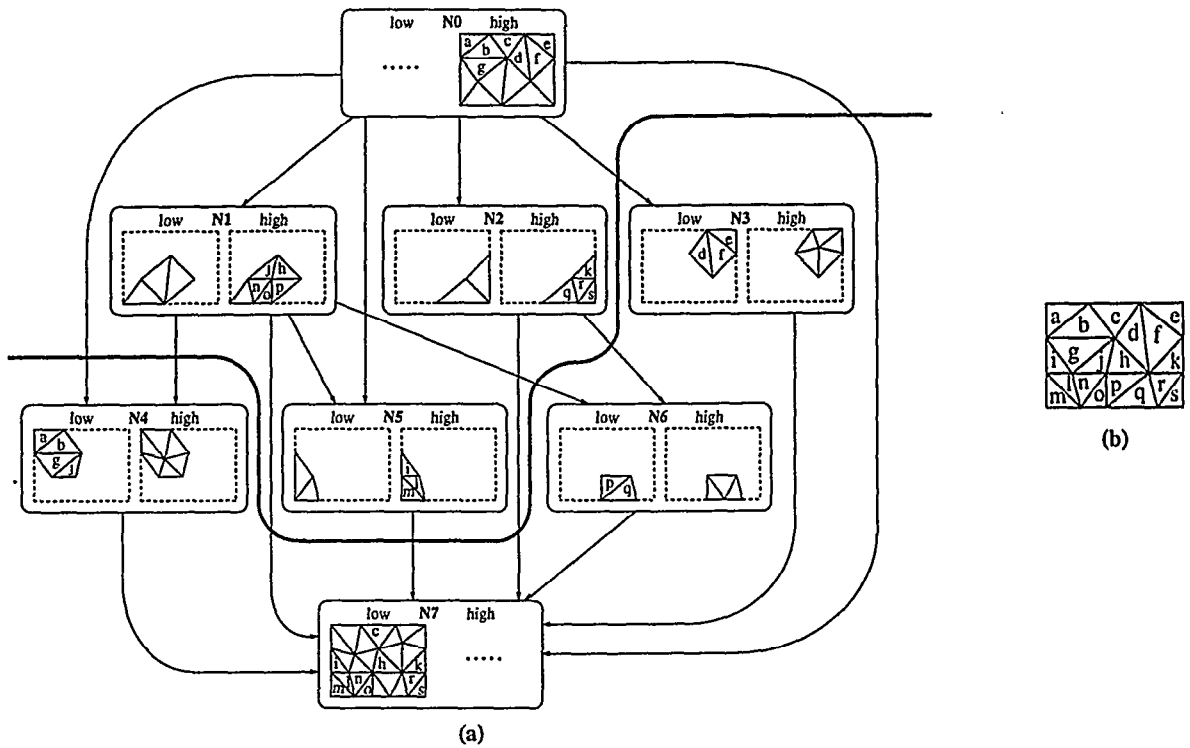


Figure 1: (a) A multi-triangulation, and a cut (shown as a thick line); (b) the variable-resolution triangulation corresponding to the cut. T triangles associated with each arc of the cut are shown.

The data structure encoding an MT [10, 4] stores components as raw nodes, while triangles are represented in the arcs of the DAG: every arc is labelled with the set of triangles associated with it. Thus, every triangle is stored exactly once, and the space complexity is linear in the number of triangles of the MT. An alternative data structure that avoids storing triangles explicitly, hence highly reducing the storage requirements of the model, is described in [4]: in this case, reduced storage is traded for less efficient query operations.

3 System Architecture

VARIANT (Variable Resolution Interactive ANalysis of Terrains) is a prototype system for the analysis, manipulation, and visualization of terrains at variable resolution, having the MT as a core model. The main idea behind *VARIANT* is to provide an open system which can be integrated with modules performing any kind of analysis and processing of terrains, with the additional characteristic of using the accuracy of representation, possibly variable over the terrain domain, as a further parameter in the computation.

The system consists of various modules, and each can be implemented as an independent process. The use of the MT model is made transparent to the applications by providing a standard interface to access it, made of a collection of operations encapsulated in a kernel module, called the *MT-manager*. The *MT-manager* is the only module having direct access to the MT data structure, and it provides three classes of primitive operations:

- *query operations*, used to inquire the MT for obtaining information on the terrain over a given area and at a given resolution;
- *update operations* used to build and modify an MT;

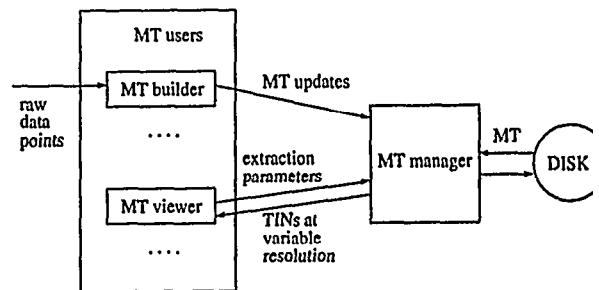


Figure 2: The architecture of *VARIANT*

- *I/O operations* used to read/write a terrain encoded by an M from/to disk.

All applications processing a terrain through the *MT-manager* are called *MT-clients*. A special *MT-client* is an *MT-builder*, i.e. program that builds the multi-triangulation off-line starting from raw dataset (e.g., a set of sample points) at high resolution. An M builder uses update operations to build the MT, and *I/O* operations to store it on disk. The system architecture is depicted in Fig 2. The *MT-manager*, some examples of *MT-builders*, and of M clients performing relevant application tasks are described in next sections.

4 The MT-manager

The *MT-manager* is essentially a layer of operations built on top the MT data structure. *Update operations* support the constructi

and modification of an MT through insertion/deletion of components. An initialization operation creates an MT corresponding to a single TIN: such an MT is made only of a source and a drain components, each holding the same TIN, and an arc connecting them. Insertion operations permit to add a new component inside the MT, by specifying the TINs it holds: such a new component is connected to other components already present in the MT, and the content of such components is modified accordingly. Deletion operations act symmetrically by eliminating a component from the MT, and updating contents and arcs on its parent and children nodes. More details on the use of update operations are given in the next section.

Most applications load a terrain model through *I/O operations*, and access it through *query operations*. Examples of basic query operations that are of interest in many applications are *point location* and *windowing*. Point location is needed whenever the characteristics of the terrain (e.g., elevation, slope) must be obtained at a given location and with a given accuracy. Windowing is a fundamental operation whenever an application must bound the domain of interest to a given rectangular portion of the domain. The MT-manager provides operations for performing these and other operations, at any desired level of accuracy, on a terrain represented through a Multi-Triangulation.

All query operations can be seen as special cases of a single general query that is specified by the following parameters:

- the maximum *error tolerance* for the inquired representation at each point of the domain: tolerance is provided by a *threshold function* $\tau : \mathbb{R}^2 \rightarrow \mathbb{R}$, where $\tau(p)$ is the error tolerance at point p .
Common examples are either uniform accuracy given by a constant threshold function, or an error increasing with distance from a reference set (e.g., a viewpoint, a set of viewpoints, a road, a coastline, a drainage network, etc.);
- a *focus set* \mathcal{F} in the plane, which defines the *area of interest* of the query. Only terrain elements interfering with \mathcal{F} are considered in extracting a representation from the MT. Interesting examples of focus sets are:
 - a point p : only the triangle containing p is reported; this corresponds to a point location query, and can be used to evaluate local terrain characteristics (e.g., elevation or slope at p);
 - a (poly)line l : used, for example, to obtain the terrain configuration along a street, a river, or a line-of-sight;
 - a region r : the extracted terrain representation is restricted to triangles lying (at least partially) inside r ; relevant cases are when r is a rectangular window, and when r is the view volume for terrain visualization;
 - the whole domain: in this case, a global representation of the terrain is obtained.

The general query consists of retrieving from the MT a terrain representation made of triangles satisfying τ and relevant with respect to \mathcal{F} . There are two different approaches to the solution of such a query:

- in a *static approach*, the answer to each query is computed independently;
- in a *dynamic approach*, the answer is obtained by modifying the solution of a previous query of the same kind.

In practice, the latter approach consists in *updating* a representation that is expected to be “close” to the desired answer. The convenience of using either one or the other approach is highly dependent on the characteristics of the application: while an application performing random point locations on the surface may prefer a static approach, a flight simulator, that must update the representation at frame rate according to the position of the viewpoint, might prefer the dynamic approach.

Given a threshold function τ , a focus set \mathcal{F} , let us call *valid* all triangles that satisfy τ , and *active* all triangles that intersect \mathcal{F} . An algorithm to answer the general query locates the smallest cut of the DAG such that all active triangles in such a cut are valid. Only such active and valid triangles are reported in the output (see Figure 3).

The static algorithm for answering the general query is essentially an adaptation of the algorithm described in [10] for extracting a representation at variable resolution over the whole terrain. The DAG is traversed starting at its root, and assuming the cut immediately below such a node as an initial solution. If some active triangle in the current cut is not valid, then the cut is moved below the end node of the arc corresponding to such a triangle. The cut is iteratively advanced until all its active triangles become valid.

In a dynamic approach, a similar strategy is applied, starting at a cut corresponding to a previous solution of a similar query (e.g., with a slightly different threshold function and/or focus set). The cut is advanced first, in order to satisfy τ at all its active triangles. However, the solution obtained in this way may be over-refined. Thus, the cut is iteratively moved towards the source of the DAG, whenever this is possible while maintaining all active triangles valid.

5 The MT-builders

MT-builders are programs that use the update operations provided by the MT-manager in order to build an MT from raw data. Input data are assumed to be in the form of a set of points sampled on the terrain surface. Such data can be obtained either directly, through data acquisition techniques (e.g., photogrammetry), or by processing other data sources, such as a digital terrain model (DTM) at high resolution, encoded either as a raster, or as a TIN, or as a contour map. A TIN containing the whole set of data points as vertices is a reference terrain model at the highest possible resolution (on the basis of the given dataset). Given the input dataset, there are two basic techniques for constructing multi-triangulations:

- *refinement* methods start from a coarse TIN and produce more refined components by adding vertices progressively in order to increase resolution;
- *simplification* methods start from the reference TIN and produce less refined components by reducing the number of vertices progressively.

In case a refinement method is adopted, the MT is initialized with a TIN built on a small set of vertices, e.g., on the vertices of the convex hull of the terrain domain. A refinement step consists in adding one component immediately above the drain of the DAG. Such a component is obtained by adding one new vertex from the dataset, and updating the drain TIN accordingly: the two local TINs held by the new component cover the portion of domain affected by insertion, before and after the vertex has been inserted, to obtain a lower and a higher resolution, respectively. The element to be inserted at each step may be selected by several heuristics, e.g., either random, or taking the point that is likely to improve the approximation better.

Simplification methods act in a fully symmetric way. The MT is initialized with a TIN at full resolution, and simplification steps consist in adding one component immediately below the source of the DAG, and updating the source TIN accordingly. In this case, the new component is obtained by removing one vertex from the source TIN, and the local TINs held by it cover the portion of domain affected by such an update, before and after deletion, respectively. Also in this case, selection of the element to be eliminated can be either random, or error-driven.

More details on the construction algorithms can be found in [6].

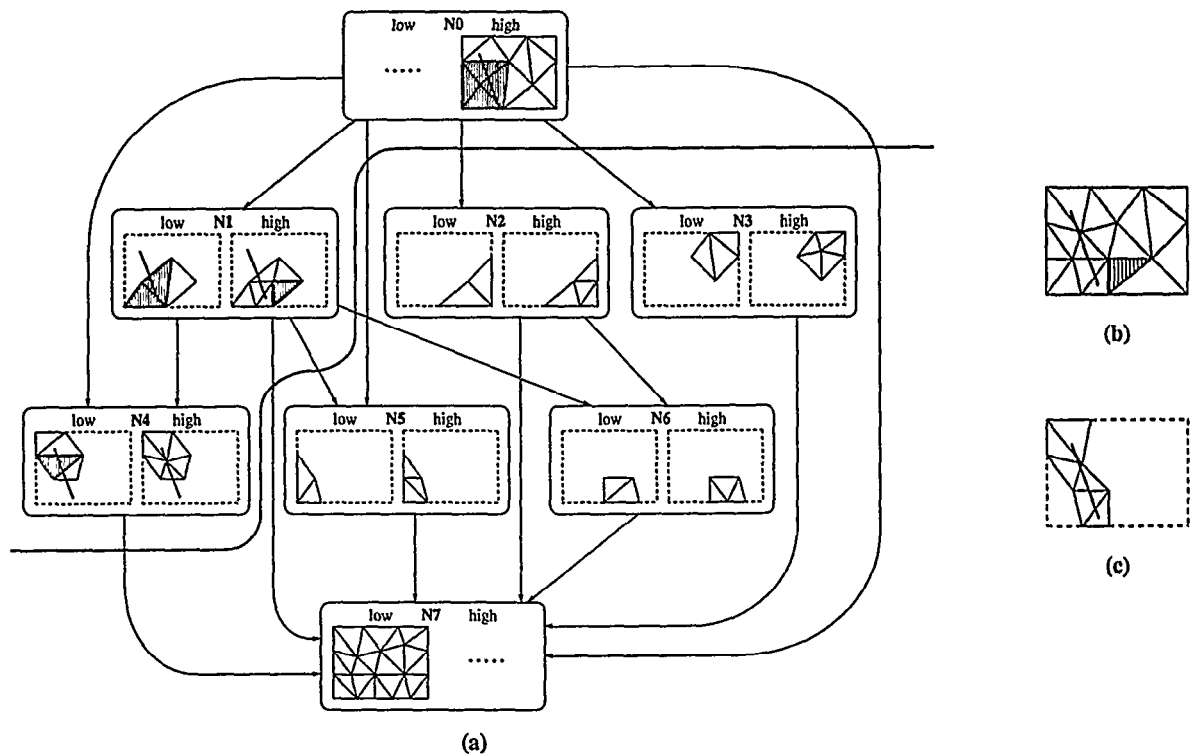


Figure 3: (a) A query on an MT (dashed triangles do not satisfy the threshold function, the thick line is the focus set), and the cut giving the solution; (b) the triangulation corresponding to such cut and (c) the output TIN; note that non-valid triangles are tolerated if they are not active.

6 The MT-clients

A naive use of MTs from any application handling terrain data (for visualization, analysis, or simulation) consists simply of extracting a TIN at a user-defined accuracy, and using it as a dataset for subsequent processing. In this case, a single TIN extraction is performed by using a constant function as error tolerance, and the whole domain as focus set. This simple feature is already an improvement over the use of a predefined terrain model at fixed resolution, since it allows an application to run on a model whose accuracy is suitable to match its needs, and/or the computing power.

However, the true power of VARIANT is exploited by applications that make direct use of variable resolution features to improve performance. In the following, we list some examples of MT-clients that either we have developed, or we are designing, which are meant to exploit such features.

Visualization. Realistic visualization of terrains is very important in flight simulation and environment assessment. The performance of a visualization system can be improved enormously if a variable accuracy of representation is used [1, 3, 8, 10].

For instance, a flight simulator can be built on top of VARIANT, which extracts at each frame a new TIN, by using a function increasing with distance from the current viewpoint as a threshold for the error tolerance, and the view frustum as a focus set. This approach permits to maintain a high visual quality, while highly reducing the number of rendered triangles.

In landscape visualization, it is possible to enhance readability of an image by using high visual quality only at a set of points or regions of interest, or along given terrain features (roads, rivers, coastlines, ridges, etc.). This is achieved simply by using suitable

threshold functions and focus sets.

Another application might simulate driving a car along a road on the terrain. In this case, it is possible to use a threshold function increasing with distance from the road, and a focus region involving only portions of terrain that are visible from the road. After an extraction with such parameters, the same (highly simplified) TIN can be used at each frame during simulation.

Drainage networks. Programs that compute drainage, and simulate behavior of water flow can take advantage of the possibility to use higher accuracy only in the proximity of most important flow channels (rivers), and/or of critical regions (e.g., regions that might be subject to floods). Also in this case, it is possible to improve performance by adopting suitable threshold functions. For instance, if the drainage network related to a given river must be extracted, the threshold function can be increasing with distance from the river, while the focus region might be limited to the slopes around its valley.

Visibility. Applications involving visibility over terrain may need either computing the viewshed from a given viewpoint, or solving problems which involve lines-of-sight between pairs of points on the surface [9].

In the former case, the focus set is the whole domain, while in the latter it is just a given line-of-sight. In both cases, the threshold function can be arbitrarily chosen depending on the needs of the specific application. In general, visibility computations are sensitive to elevation errors near the viewpoint, therefore it is recommended to vary resolution according to the distance from the viewpoint [7], adopting a highly accurate representation near the viewpoint and increasing the error threshold while moving away from it.

Note that a query on the mutual visibility of two points is implicitly answered by using the line-of-sight as focus set: therefore, a visibility program that manipulates only lines-of-sight does not need to load directly a terrain model, which is handled indirectly through the MT-manager.

7 The Prototype System

VARIANT has been implemented in C++ under the Unix operating system and the X-Window environment; its interactive user interface has been developed by using the FL-FORMS and OpenGL graphics libraries. The system is portable to any Unix platform, and it currently runs on Silicon Graphics workstations, and PCs (under Linux).

In the current state, the following modules have been implemented:

- Two MT-builders, which construct an MT through refinement and simplification techniques, respectively, based on the update of a Delaunay triangulation by incremental insertion or deletion of points. Such construction algorithms are based on well-known procedures to dynamically update a Delaunay triangulation. Other MT-builders, based on data dependent triangulations [11] rather than on the Delaunay triangulation, are under implementation.
- Two MT-managers, using two alternative data structures for storing an MT: a highly efficient data structure where triangles are represented explicitly, and a lighter data structure, ad-hoc for Delaunay TINs, which encodes only the vertices of the model (see Section 2). This latter structure only implements the static algorithm for answering the general query. There is trade-off between the efficiency of the data structure and the efficiency of the extraction algorithm. Results and comparisons can be found in [4].
- An MT-client, which permits the interactive visualization of a terrain at variable resolution. Two viewing modes are supported:
 - In *overview mode*, the focus set is the whole domain, and the user can select different threshold functions to extract a TIN at variable resolution. The terrain initially appears centered in the main window; then the user can rotate, translate and zoom on it.
 - In *animation mode*, the user can fly on the terrain by driving its viewpoint as a virtual “airplane”. Controls similar to those of a flight simulator are provided. The threshold function linearly increases with distance from the current viewpoint, and therefore changes at every instant of time as the viewpoint moves.In both viewing modes, colors, lights and rendering parameters can be edited to obtain the desired image appearance.

8 Conclusions and Future Work

VARIANT is a prototype system for processing and visualizing terrains at variable resolution, based on the Multi-Triangulation. For its modular structure, the system is “open” to the addition of further modules. We are currently enriching the system with improved extraction algorithms which also generate *topological information* for the output TIN. In fact, an output simply consisting of a collection of triangles is enough for visualization, but not for more complex terrain analysis, such as visibility computations, construction of drainage networks or basins (see Section 6). We also plan to add *more MT user modules*, performing specific computations such as viewshed computations, extraction of drainage networks and basins, computation of optimal paths, etc. Finally, we are

studying an *indexing technique* for multi-triangulations. In fact, the size of an MT for large data sets could be too big to fit in main memory, and in such cases methods are needed that allow the efficient retrieval of the disk page containing a part of an MT, when access to that portion is required.

Acknowledgements

This work has been partially supported by the BRITE Basic Research Project *VENICE (Virtual Environment Interface by Sensory Integration for Inspection and Manipulation Control in Multifunctional Underwater Vehicles)* and by the Coordinated Projects of the Italian National Research Council *Models and Systems for Handling Environmental and Territorial Data* and *Multiresolution models for visualization of multidimensional fields*. We wish to thank Roberto Sagoleo, Tiziana Vignola, Paola Caluzzi, Mauro Larosa, Luca Landi, Anna Anacleto, and Ivan Nicora who worked at the implementation of the system.

References

- [1] P. Cignoni, E. Puppo, R. Scopigno, 1997 Representation and visualization of terrain surfaces at variable resolution, *The Visual Computer*, 13 (in press).
- [2] M. de Berg, M., K.T.G. Dobrindt, 1995, On the levels of detail in terrains, *Proc. 11th ACM Symp. on Computational Geometry*, Vancouver, Canada, c26-c27.
- [3] L. De Floriani, P. Magillo, E. Puppo, M. Bertolotto, 1996, Variable Resolution Operators on a Multiresolution Terrain Model, *Proc. 4th ACM Workshop on Advances in Geographic Information Systems (ACM-GIS '96)*, Rockville, MD, USA, November 1996.
- [4] L. De Floriani, P. Magillo, E. Puppo, 1997, Efficient encoding and retrieval of triangle meshes at variable resolution, *Technical Report N. PDISI-97-01*, DISI – University of Genova, Italy, 1997.
- [5] L. De Floriani, P. Magillo, E. Puppo, 1997, VARIANT - Processing and Visualizing Terrains at Variable Resolution, *Technical Report N. PDISI-97-05*, DISI – University of Genova, Italy, 1997.
- [6] L. De Floriani, P. Magillo, E. Puppo, 1997, Building and traversing a terrain at variable resolution, *Technical Report N. PDISI-97-02*, DISI – University of Genova, Italy, 1997.
- [7] J.P. Felleman, C. Griffin, 1990, The role of error in GIS-based viewshed determination - a problem analysis, *Technical Report EIPP-90-2*, Institute for Environmental Policy and Planning, State University of New York.
- [8] P. Lindstrom, D. Koller, W. Ribarsky, L.F. Hodges, N. Faust, G.A. Turner, 1996, Real-time, continuous level of detail rendering of height fields, *Proc. SIGGRAPH 96*, in *Computer Graphics - Annual Conference Series*, ACM SIGGRAPH, New Orleans, LA, USA, August 1996, 109-118.
- [9] G. Nagy, 1994, Terrain visibility, *Computers and Graphics*, 18, 6, 763-773.
- [10] E. Puppo, 1996, Variable resolution terrain surfaces, *Proc. 8th Canadian Conf. on Computational Geometry*, Ottawa, Canada, August 1996, 202-210. Longer version available as *Technical Report N.6/96*, IMA – C.N.R., Genova, Italy, 1996.
- [11] S. Rippa, 1992, Adaptive approximations by piecewise linear polynomials on triangulations of subsets of scattered data, *SIAM Journal on Scientific and Statistic Computing*, 13, 1, 1123-1141.
- [12] H. Samet, 1990, *Applications of Spatial Data Structures*, Addison Wesley, Reading, MA.