A2-21
2000

Volume Visualization of Large Tetrahedral Meshes on Low Cost Platforms

Paolo Cignoni[1], Leila De Floriani[2], Paola Magillo[2], Enrico Puppo[2], Roberto Scopigno[3]

[1] Istituto di Elaborazione dell'Informazione Consiglio Nazionale delle Ricerche*

[2] Dipartimento di Informatica e Scienze dell'Informazione Università di Genova†

[3] Istituto CNUCE Consiglio Nazionale delle Ricerche‡

Abstract

In this paper, we present an interactive system for visualization of three-dimensional scalar fields. The system has been designed in order to overcome some of the problems posed by very large volume data sets. The adopted solution exploits an efficient multiresolution data structure based on a tetrahedral domain decomposition. The mesh to be rendered can be selectively refined over areas that the user considers more critical, on the basis of either field values or domain locations. The system supports different rendering techniques, such as isosurface fitting and Direct Volume Rendering (DVR).

1 Introduction

Space requirements and rendering time for a tetrahedral mesh are proportional to its resolution, i.e., to the density of its tetrahedra, and, thus, to the density of the underlying point set. When a tetrahedral mesh is too large, data simplification can be used in order to bring it to a more manageable size prior to visualization [3, 1, 8, 7, 6, 11].

In many cases, it is very useful to apply simplification selectively, e.g., only in a portion of the domain, or only in the proximity of interesting field values. This process is usually called *selective refinement*.

Mesh simplification, however, is too time-consuming to be performed on line. A more recent approach consists of decoupling the simplification phase from the selective refinement phase. This has led to multiresolution models, which have been developed in the two-dimensional case for triangle meshes (see, e.g., a survey in [5]). Basically, such models encode the simplification steps as a partial order, from which a virtually continuous set of meshes at different Levels Of Detail (LODs) can be extracted, based on efficient graph traversal strategies.

Here, we present a volume visualization system, called *TAn2* (*Tetrahedra Analyzer* 2), which is based on a multiresolution model which specializes for the three dimensional case the model proposed in [4, 9], called the *Multi-Tesselation* (MT).

*CNR Research Area – S. Cataldo 56100 Pisa, ITALY – Email: cignoni@iei.pi.cnr.it

†Via Dodecaneso, 35, 16146 Genova, ITALY – Email: {deflo,magillo,puppo}@disi.unige.it

‡CNR Research Area S. Cataldo 56100 Pisa, ITALY Email: roberto.scopigno@cnuce.cnr.it

TAn2 exploits multiresolution not only for reducing data complexity uniformly, but also for varying the resolution through data domain. Selective refinement can be based either on the levels of the isosurfaces to be rendered, or on a spatial location within the domain. These features allow the user to focus on those areas that are considered more critical.

One of the major objectives of *TAn2* is to enable a low cost PC platform to perform those tasks that are nowadays the realm of more costly graphics workstations. We assume a PC architecture equipped with 512MB RAM and a graphics board having a sustained throughput at least of 1M shaded triangles/sec. We consider an application which manages datasets based on a *tetrahedral decomposition* of the 3D space, with scalar data associated with the mesh geometry, and supports Direct volume Rendering (DVR), isosurface fitting and cross-slice rendering at a reasonable degree of interactivity (10 fps). Under the above assumptions, a standard volume visualization system could only work with a mesh not exceeding either 350K vertices when doing isosurface rendering, or 3K vertices when doing DVR (see [2] for details). On the contrary, *TAn2* can manage meshes of up to 12M vertices by using user-driven selective refinement prior to visualization. A more detailed description, presenting also the underlying data structure and the selective refinement techniques on which *TAn2* is based, can be found in [2].

2 The *TAn2* system

The architecture of the *TAn2* (Tetrahedra Analyzer 2) system has been designed to overcome some of the problems arising interactive visualizing very large volume data sets. The solution adopted in *TAn2* is based on a multiresolution structure that describes the whole data set in an implicit and very compact way, and supports selective refinement efficiently [2]. From such structure, smaller subsets of data are extracted and rendered. The amount of data extracted for rendering and their level of resolution (possibly variable over the domain) are selected in such a way to give the largest mesh that can be rendered under the given time and resource constraints. The system has been implemented in C++ and runs under both Windows and Linux.

TAn2 system provides both visualization through isosurfaces, cross sections and Direct Volume Rendering (DVR) on *uniform* or *variable LOD*. In isosurface rendering mode, one or more isosurfaces may be displayed at a time; data visualized can be enriched by introducing cross section planes (see Figure 3). The DVR approach adopted is based on a depth-sorting and a composition of tetrahedral cells [10, 12].

In all rendering modalities (isosurfaces, cross planes and DVR), appearance parameters such as colors and transparency values are determined by means of a user-defined *Transfer Function* (TF). The Transfer Function defines how field values are mapped into color and transparency values that will be used in rendering. The TF is defined in terms of RGBA components through linear interpolation of the RGBA values specified by the user at a finite set of field values. The system contains a visual interactive tool for editing transfer functions (see Figure 2).

2.1 Role of Multiresolution within the System

Within the system, a compact *multiresolution data structure*, the Multi-Tesselation (MT), is maintained, which describes the whole data set in an implicit way. This data structure requires only about $36n$ bytes for storing a model having n vertices at the highest resolution, and for supporting all multiresolution operations. In [2], we show that the MT has a compression factor of about 6 with respect to storing just the mesh at the maximum resolution with a standard data structure.

For the purpose of rendering, a *current mesh* is stored, which is dynamically extracted from the MT by selecting just a subset of the data. Rendering is performed based on the current mesh. The current mesh has a suitable size to be rendered with the available memory and with an interactive frame rate, depending on the current viewing modality. The resolution of the current mesh is either uniform or variable over the domain, and is chosen based on the current viewing parameters so as to provide the highest possible quality of displayed images within the size constraints. In particular, the system supports the following uses of multiresolution:

- *Uniform LOD*: the resolution of the current mesh used in rendering is uniform over the domain based on an approximation error threshold specified by the user.

- *Variable LOD based on spatial location in the domain*: the user specifies both an error threshold and a focus volume, e.g., an axis-aligned box. The focus volume is used as a sort of 3D magnifying glass in order to explore a dataset by increasing resolution only locally.

- *Variable LOD based on field value*: the current mesh satisfies the error threshold specified by the user just in proximity of the isosurface values currently selected, while the resolution of the mesh in areas that do not contain such values is coarser.

TAn2 accepts as input either a plain tetrahedral mesh or a Multi-Tesselation. The construction of the Multi-Tesselation is performed off-line by an independent application; we use a simplification code based on edge collapse described in [1].

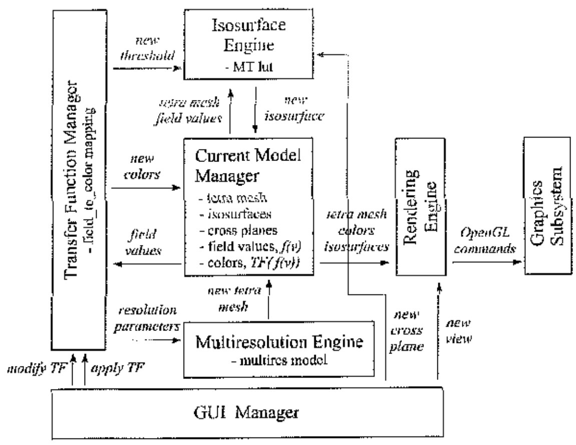

Figure 1: The architecture of the *TAn2* system.

3 System Architecture

The architecture of the *TAn2* system is depicted in Figure 1 and consists of six modules which are described in the following paragraphs. The **GUI Manager** is responsible for rendering all GUI-related areas of the system interface, managing user interaction, and filtering all user inputs before passing them to other modules.

The interface of *TAn2*, shown in Figure 2, consists of one main window subdivided into two regions. The *View region* (on the left) is devoted to the visualization of the dataset. The 3D scene visualized in the View region is managed by the *Rendering Engine*. The *Transfer Function region* (on the right) gives an innovative unified GUI framework for the interactive setting of the current TF, the selection of isosurface thresholds, and the selection of the error threshold for mesh extraction from the MT.

The *Transfer Function region* contains a visual editor to interactively design or modify the current TF, to set up the parameters for the selection of isosurfaces and for the extraction of the current mesh from the MT. Changes to either the TF, or the isosurface values, or the error threshold, are managed by the *Transfer Function Manager* with the help of the *Multiresolution Engine* and of the *Current Model Manager*.
The **Transfer Function Manager** collects all parameters set by the user, which are related to the current TF, to the current isosurface values, and to the current error threshold. The Transfer Function Manager acts as a filter that redirects the user specifications to the appropriate modules which handle them.
The **Multiresolution Engine** is driven by interactively manipulating either the handle of the current error threshold in the Transfer Function region of the GUI, or the *manipulator* 3D widget provided to select the spatial focus region. The *GUI Manager* communicates any changes to the *Transfer Function Manager*, which in turn invokes the *Multiresolution Engine*. Based on the new error threshold, the *Multiresolution Engine* extracts a new mesh from the MT and sends it to the *Current Model Manager*.

The **Current Model Manager** is devoted to the management of the current mesh: a dynamically modifiable tetrahedral mesh that stores the model as extracted from the MT, enriched with data needed in various visualization modalities (like isosurfaces, colors, etc), in other words, it maintains the *ready-to-render* model.
The **Isosurface Engine** computes new isosurfaces on command of the *Transfer Function Manager*, or new cross sections following the requests of the *GUI Manager*. These updated surface meshes are maintained in the *Current Model Manager*.
The **Rendering Engine** performs rendering either of isosurfaces, or directly of the data set (possibly through DVR).

In the example of Figure 2, a uniform resolution mesh has been extracted (the corresponding error threshold is the one selected with the multiresolution extraction handle), and three different isosurfaces are fitted and rendered (corresponding to the three isosurface handles shown) together with the mesh boundary surface.

4 Results

In this section, we show images and performance statistics obtained from the *TAn2* system on the **Turbine Blade** dataset (106,795 vertexes, 576,576 tetrahedra), a curvilinear mesh courtesy of AVS Inc. (tetrahedralized by O.G. Staadt). Turbine Blade is a rather large dataset, non-convex, containing many degenerate cells (i.e., having zero volume). It contains 41,920 boundary cells. An estimate of the average size of the isosurfaces fitted on this data set has been computed by considering nine field values equally distributed in the dataset field range: the estimated average size is about 70K faces per isosurface.

Meshes at uniform resolution have been extracted from the multiresolution representation of the Turbine dataset according to a constant error threshold. The sizes of such meshes are reported in Table 1, where the error value is measured as a percentage of the field range. Table 1 contains also results on isosurface generation from *variable LOD* meshes constructed on the basis of the field value. The value of the resolution threshold used in mesh extraction ($\varepsilon = 0., 0.1$ and 1.0) are measured as percentages of the dataset field range; the corresponding mesh sizes are also reported. It has to be noted that if the dataset contains an isosurface which intersects a large number of mesh cells, then the corresponding *variable LOD* cannot be very concise. An example is provided by the *variable LOD* mesh corresponding to the active isosurface of value 160.28 and to an error $\varepsilon = 0.$ (see Table 1). In this case, the joint use of a *field*-based and of a *spatial domain*-based focusing filter can improve data reduction capabilities. While under the configuration described above ($\varepsilon = 0.$, isosurface=160.28) we obtain a *variable LOD* mesh composed of 473K cells, we can easily produce a representation of smaller size (in the range of 80K cells) by focusing on a domain subvolume. The multiresolution extraction time is in general nearly interactive.

Finally, images representing the other two rendering modalities are in Figures 3 and 5. In the first one, we show a cross plane extracted using a *variable LOD* (40k cells out of the original 576k) under a resolution error threshold $\varepsilon = 2\%$. In the latter, we show a hybrid DVR image (integrated isosurfaces plus DVR) on the same *variable LOD*.

5 Conclusions

The *TAn2* system supports rendering of datasets that are about six times larger than those that could fit in memory with a traditional system, and from 30 to 4000 times larger than those that could be visualized interactively (depending on whether we are doing isosurface rendering or DVR).

To our knowledge, *TAn2* is the only multiresolution volume visualization system able to deal with irregular datasets and non-convex domains efficiently.

References

[1] P. Cignoni, D. Costanza, C. Montani, C. Rocchini, and R. Scopigno. Simplification of tetrahedral volume with accurate error evaluation. In *Proceedings IEEE Visualization 2000*, page in press. ACM Press, October 7-13 2000.

[2] P. Cignoni, L. De Floriani, P. Magillo, E. Puppo, and R. Scopigno. *TAn2* - visualization of large irregular volume datasets. Technical Report DISI-TR-00-07, DISI, University of Genova (Italy), 2000. (submitted paper).

[3] P. Cignoni, L. De Floriani, C. Montani, E. Puppo, and R. Scopigno. Multiresolution modeling and rendering of volume data based on simplicial complexes. In *Proceedings 1994 Symposium on Volume Visualization*, pages 19–26. ACM Press, October 17-18 1994.

[4] L. De Floriani, P. Magillo, and E. Puppo. Building and traversing a surface at variable resolution. In *Proceedings IEEE Visualization 97*, pages 103 110, Phoenix, AZ (USA), October 1997.

[5] M. Garland. Multiresolution modeling: Survey & future opportunities. In *Eurographics '99 State of the Art Reports*, pages 111–131, 1999.

[6] M.H. Gross and O.G. Staadt. Progressive tetrahedralizations. In *Proceedings IEEE Visualization'98*, pages 397 402, Research Triangle Park, NC, 1998. IEEE Comp. Soc. Press.

[7] R. Grosso and G. Greiner. Hierarchical meshes for volume data. In *Proceedings CGI'98*, Hannover, Germany, June 22-26 1998.

[8] J. Popovic and H. Hoppe. Progressive simplicial complexes. In *ACM Computer Graphics Proceedings, Annual Conference Series, (SIGGRAPH '97)*, pages 217–224, 1997.

[9] E. Puppo. Variable resolution terrain surfaces. In *Proceedings Eight Canadian Conference on Computational Geometry*, pages 202 210, Ottawa, Canada, August 12-15 1996.

[10] P. Shirley and A. Tuchman. A polygonal approximation to direct scalar volume rendering. *Computer Graphics (San Diego Workshop on Volume Visualization)*, 24(5):63 70, November 1990.

[11] I.J. Trotts, B. Hamann, and K.I. Joy. Simplification of tetrahedral meshes with error bounds. *IEEE Transactions on Visualization and Computer Graphics*, 5(3):224–237, 1999.

[12] P.L. Williams. Visibility ordering of meshed polyhedra. *ACM Transaction on Graphics*, 11(2):103 126, April 1992.

Turbine variable resolution models										uniform resolution models	
isosurface threshold	ε	tetra	isosurf. size	ε	tetra	isosurf. size	ε	tetra	isosurf. size	ε % of field range	
137.38	0.	352,556	73,952	0.1	134,901	33,759	1.	40,611	16,296	0.1	233,848
160.28	0.	473,625	141,978	0.1	169,824	67,909	1.	46,048	24,017	0.5	117,696
183.17	0.	263,311	28,718	0.1	100,798	18,081	1.	36,148	8,343	1.0	84,743
206.07	0.	190,988	8,296	0.1	69,604	6,858	1.	28,637	3,167	5.0	37,515
217.52	0.	108,298	3,001	0.1	47,810	2,621	1.	22,486	1,548	10.0	23,779

Table 1: Results of generating isosurfaces through the extraction of meshes at *variable* resolution from the multiresolution representation of the Turbine dataset. The error is measured as a percentage of the field range, and depends on the current selected isosurfaces (it is $\leq \varepsilon$ on tetrahedra contributing to the isosurface cells; any error on other tetrahedra). Last two columns show sizes of meshes at *uniform* resolution extracted from the multiresolution representation of the Turbine dataset.

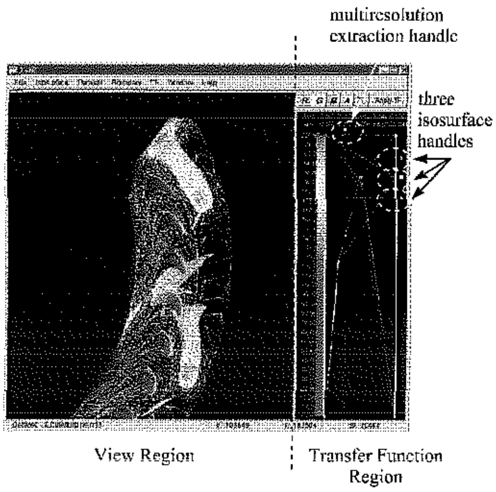

Figure 2: The main window of the *TAn2* system. The isosurfaces shown have been computed on a *uniform LOD* mesh (containing 183K tetrahedra) extracted from the multiresolution representation of the Turbine dataset.

Figure 3: An image of a Turbine *variable LOD* (40k cells, $\varepsilon = 2\%$) with a cross plane.

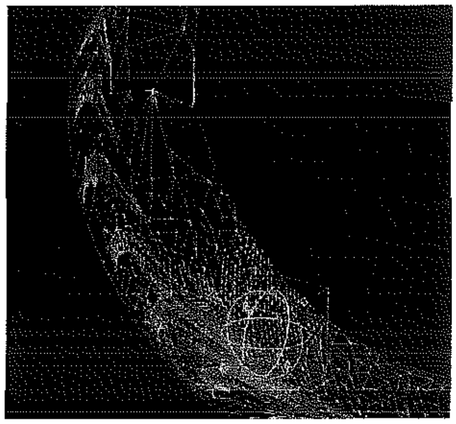

Figure 4: An image of a Turbine *variable LOD* mesh (65k cells, extracted under approximation error $\varepsilon = 0.01\%$ in the interior of the focus region, and error $\varepsilon = 10\%$ in the external space).

Figure 5: An hybrid DVR (integrated isosurfaces plus DVR) image of a Turbine *variable LOD* (40k cells, $\varepsilon = 2\%$) is shown.