Distinctions Of The Multiresolution Model Of Surface For Regularly Distributed Data

A. Kapustin, A. Kryachko, J. Fedorova, T. Firsova
Nizhny Novgorod Software Technology Lab
Nizhny Novgorod, Russia

Abstract

During the last years the conception of the multiresolution model (MRM) of surface was developed for representation of the geometric objects with continuous level of detail (LOD). The basic idea of this model is the storing of the triangulations with different LODs in the directed acyclic graph (DAG) structure. The model is created at the preprocessing step and provides extracting of the consistent triangulation with continuous LOD from DAG in real time.

The sample algorithms for the creation of MRM for arbitrary surface triangulation are described in [1]. The algorithm for DAG creation is time and memory consuming.

The regularity of data allows to increase the performance of MRM significantly. In that case no complex refinement/decimation algorithms are needed for DAG creation. Memory overhead is also decreased compared with the arbitrary data memory overhead. Also, we managed to represent extracted triangulation as a single generalized triangle-strip.

We used this approach of DAG creation and LOD extraction in two applications: one for real-time rendering of the regular terrain and the other for adaptive representation of the surface in 3DStudioMax footprint simulation plug-in.

The algorithm may be used in different areas where significant reduction of primitives without any noticeable loss of quality is required, e.g., computer graphics (real-time rendering, games), GIS or physical modeling (intensive calculations on meshes with huge number of primitives), etc.

Keywords: *multiresolution model, multi-triangulation, level of details, directed acyclic graph.*

INTRODUCTION

The concept of multiresolution models (MRM) was developed in the recent years to solve the problem of representing geometry objects with variable resolution. Multiresolution model ensures a compact description of a large number of triangulations that represent a given object with variable resolutions. The model that is built at the preprocessing stage allows to extract consistent triangulation of different LOD in real time, i.e., triangulation whose resolution is evenly spread over the surface. MRM enables to render the surface at any time with minimal LOD needed to reach the quality required. Multiresolution geometry models support representation and processing of geometry objects on different LOD. These models help establish efficient control over geometry data in accordance with the specific application requirements. The case of one-to-one projection of surfaces on the plane is of special interest because of the GIS applications where the surfaces are analyzed and processed in the context of computationally intensive problems as well as the virtual reality case where the surfaces a visualized in real time. For instance, *view-dependent* terrain visualization could be accelerated by supplying to each frame gradually decreasing resolution while zooming out.

Building MRM for randomly distributed data was considered earlier in [2]. This article addresses the following problems:

a) Implementing MRM building algorithms and extracting triangulations of different LOD in real time for regularly distributed data;

b) Performance assessment against MRM for randomly distributed data;

c) Generalizing algorithms for general regularly distributed data. We have created a special module called *Adaptive Mesh* to build MRM for general regularly distributed data (attributed two-dimensional regular meshes).

The article structure is as follows: section 1 discusses other works on MRM, section 2 describes the MRM building algorithm for regularly distributed data, section 3 considers generalized algorithm for general regularly distributed data (module *Adaptive Mesh*). Section four, finally provides examples of using the algorithm in two applications.

1. RELATED WORKS

[3] offers a review of works dedicated to building MRM and extracting different LOD triangulation [3].

In [4] E. Puppo introduces the notion of multi-triangulation (MT) of surface. Multi-triangulation is defined as multiresolution surface model based on a set of fragments of flat triangulations placed in a partially ordered set. Combining different model fragments produces various surface approximations. The partial order could be represented by a directed acyclic graph (DAG). Formal definition and detailed description of MT properties and characteristics could be found in [1]. L.De Floriani, P. Magillo, and E. Puppo in [5] extended the notion of multi-triangulation for free-form surfaces.

The first MRM created on regular subdivisions were models that used quadtrees. A quadtree is a data structure to describe uniquely projected surface [6]. The initial square domain is partitioned into a set of nested squares by recursive splitting of each square into four quadrants. The main disadvantage of this models is that extracted triangulation is generally non-conforming. (In conforming triangulation any two triangles either have no common points or have one common point or have one common edge). A structure called *restricted quadtree* [7] was then introduced to overcome this problem. A restricted quadtree is a quadtree where the difference between adjacent leaves may not be greater than one level. The leaves form consistent triangulation which is achieved through triangulation of each quadrant in the following fashion: the quadrant is broken into four right triangles by diagonals; each of the four resulting triangles could be subsequently subdivided into two *right* triangles by joining the

middle of hypotenuse with the opposite vertex if the level of the quadrant adjacent to hypotenuse from the opposite side is one step higher. *Restricted quadtree* is built in three stages [8]. At the first stage the quadtree is formed, at the second the quadtree is balanced locally, i.e., the requirement for the level difference between adjacent leaves to be not greater than one is satisfied, and at the third each quadrant is triangulated. The disadvantage of this structure is that each time triangulation is extracted from the quadtree balancing and triangulating procedures must be performed. Besides, resulting triangles are not a part of the original model.

To avoid this full hierarchy formed by the right triangles must be considered. In this case the original triangulation is the initial square area triangulated by diagonals. Further, each right triangle is split into two right triangles by joining the middle of the triangle hypotenuse with the opposite vertex. The obvious advantage of that structure is the compactness, i.e., each triangle could be assigned a number by which the triangle geometry, adjacent intersecting triangles may be restored. The use of this type of hierarchy to support different resolution was first suggested in [9] for terrain visualization.

We have used the same hierarchy to build a multiresolution model.

2. ALGORITHM FOR BUILDING MRM FOR REGULARLY DISTRIBUTED DATA

Building MRM for randomly distributed data (building algorithms, DAG characteristics, ways to extract triangulation with the help of DAG, etc.) was previously described in [2]. Generally we have kept the MRM building concept in the case of regularly distributed data.

The algorithm input data are a regular height field, a set of points projected on the plane OXY in the nodes of the square regular mesh (fig.1). The start square is the square in the plane OXY with the minimal length of the side; projections of all initial points are located inside the square and on the square boundaries.

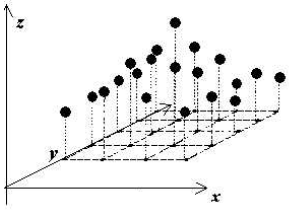


Fig.1 Regular Height Field

DAG is used to represent multi-triangulation. The graph nodes store information on local updates of the surface, arcs store information on triangles engaged in the updates. DAG is built by the top-down method starting off with the coarsest triangulation that results from subdividing the start square by the diagonal. New nodes are added to DAG during local mesh updates. A local mesh update is the subdividing of the current right triangle by the bisector of the right angle into two similar right triangles (fig.2). At this stage a new node is introduced into the mesh. The subdividing process is then repeated recursively until the maximum level of detail is reached where all vertices of the given regular set have been engaged. Fig.3 shows the resulting DAG.

DAG cut (a set of arcs) that defines surface approximation is built to extract triangulation. The surface is naturally presented as triangulation. The DAG cut is built on the basis of the given threshold function of error $\tau(P)$: $R3 \rightarrow R$, that in each point P sets the approximation error threshold in this particular point. For each triangle t the error value is computed as max(dist(V_i, t)), where dist(V_i, t) is the distance in the Z-direction from the vertex V_i to the plane of the triangle t. The pass will include all of the vertices V_i inside the triangle t projected on the plane OXY. The error value defines the quality of the surface approximation, the lower the error value, the better the approximation quality. The triangle t is included in the cut when and only when the error is less than min ($\tau(P)$) over all $P \in t$. Otherwise, triangles that result from subdividing the triangle t will be considered.

Due to regularity of local updates the DAG structure is relatively simple but not regular because of the existing requirement for the consistency of extracted triangulation. For example, when triangle 6 is subdivided (fig.2), triangle 13 must be also subdivided, otherwise consistency will be compromised. Consequently, arcs 6 and 13 must be directed to the same node.

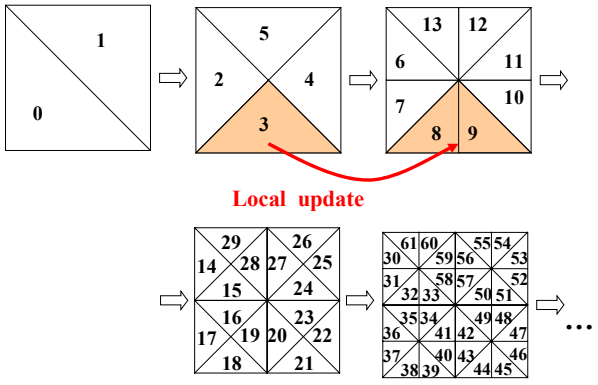


Fig.2 Building DAG.

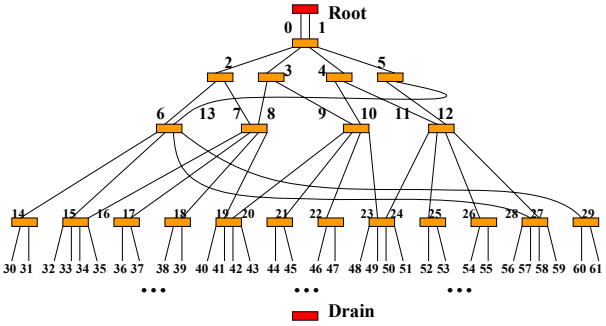


Fig.3 DAG Structure.

Additional advantages of regular local updates over irregular data include

- Significant simplification of DAG building algorithms and triangulation extraction. Due to the update simplicity no complex triangulation decimation or refinement algorithms to define the DAG structure for irregular data are required;
- Data structures are much more compact compared against the irregular data, which results in significant (more than four-fold) decrease of the memory overhead needed to store DAG. In this case the triangle hierarchy is implicit defined on the basis of the predefined patterns, there is no need to explicitly copy geometry or topology information.. Each triangle's index could be treated as a code, applying algebraic manipulations to retrieve its geometry and contiguity.

- Significant increase of the triangulation extraction algorithm efficiency due to the capability of getting most triangle information from the triangle index. Fig.4 shows the time needed triangulation extraction for regular and irregular data. Apparently, in the case of regular data triangulation is extracted several times faster and the time of extraction increases much slower as the number of triangles goes up.
- Triangulation that corresponds to any DAG cut could be represented as a single triangle strip (fig.5) to speed-up visualization. In this case triangulation may contain triangles of different LOD: the strip update algorithm is evident at each triangle subdividing.

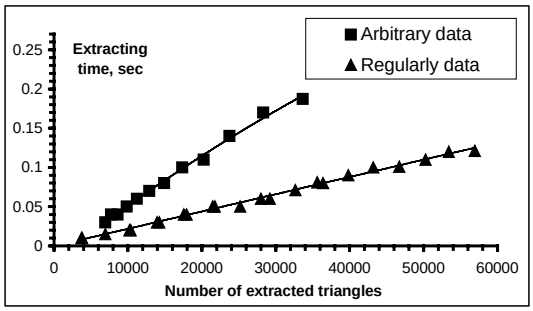


Fig.4 Triangulation Extraction Algorithm: Speed vs. Number of Extracted Triangles (Pentium II 400 MHz, 192 MB RAM)**.**

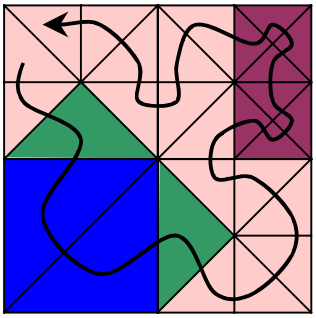


Fig.5 Representing Extracted Triangulation as One Strip. Different LOD triangles are marked with different colors.

3. BUILDING MRM FOR GENERAL REGULAR DATA

In the algorithm described above the only characteristics of the triangle vertices are the vertex coordinates. In most cases this information will suffice to compute the threshold error function. However, a number of cases will require to set the surface LOD on the basis of criteria defined by the developer. Physical modeling where local mesh refinement is necessary in at least some areas to make computations more accurate is one good example.

To achieve this objective we have created a special object, implemented as a C++ class, called Adaptive Mesh (AM). The object is intended to work with attributed regular two-dimensional meshes with the main purpose of building variable resolution triangulation of the surface on the basis of the attribute values. (adaptive surface representation).

Two-dimensional regular mesh is the basis for AM. The developer has the option of setting attribute information in the mesh nodes while during the process of working with AM. The number and type of attributes are optional except that for each mesh node the number of attributes must be the same.

Attribute values are set by the developer in the process of working with AM. The information on the attribute values is used to determine the surface LOD, namely, to compute the error in the triangle. The developer sets the method of the triangle error calculation and the threshold error value that determines admissibility for the triangle to be present in triangulation. Since the attributes are arbitrary, AM could be used in various areas, e.g., computer graphics, physical modeling, etc. In particular, setting the z-coordinate as the given point attribute makes working with AM virtually the same as with terrain.

4. EXAMPLES OF THE USE OF THE ALGORITHM

We used the algorithm and AM in two applications..

4.1 Surface Adaptive Representation in the Footprint Simulation Plug-in for 3DStudioMax

We have developed [10] a plug-in for 3DStudioMax to simulate footprints left on the surface by a moving object. The surface is represented as a regular vertex field. Whenever the cell height values are computed, the surface must be triangulated for subsequent rendering in the 3DstudioMax scene.

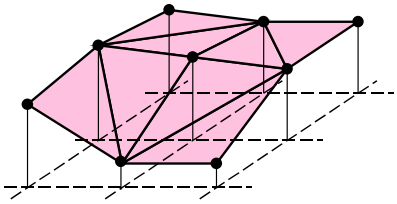
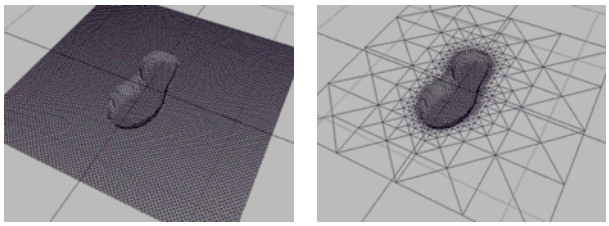


Fig.6 Primitive Triangulation of the Height Field

In the case of primitive triangulation of the surface (fig.6) the mesh detail will be apparently excessive in non-affected areas (fig.7a) which will result in unnecessary memory overhead and module performance slowdown due to significant time expense (proportional to the number of triangles in triangulation) needed to re-render the mesh in the 3DstudioMax scene.



a) *b*)

Fig.7 Height Field Triangulation: *a*) without AM (32,768 triangles). All triangles have minimally admissible size, triangulation corresponds to triangulation on fig.6. *b*) with AM (3,318 triangles). In this case the size of the triangles is minimal in the center of the surface and equals to the size of triangles in *a*). The size of triangles gradually increases from the center to the surface periphery.

The module used adaptive surface representation created by the algorithm in order to decrease excessive density of triangles in non-affected areas of the mesh(fig. 7*b*), which made the use of AM necessary because of the special surface LOD requirements.

The triangulation detail should be minimal in non-affected areas of the mesh and increase in the areas of impact. Thus, in AM we defined the error as the maximum deviation of the triangle vertices relative to the non-affected surface level (fig.8).

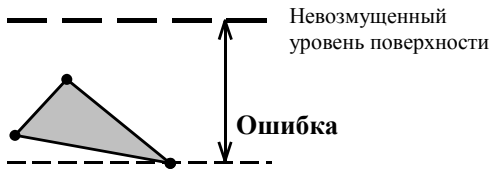


Fig.8 Error Calculation Method Used in the Module

The use of AM resulted in a significant improvement of efficiency and effectiveness of the module:

- Mesh rendering time in the 3DStudioMax scene has become much less;
- Memory overhead needed to save the mesh in 3DStudioMax has decreased. The actual memory volume becomes proportional to the size of the acting object and not the mesh size; the decrease of the memory overhead is very significant, e.g., for the mesh shown on fig. 7 the memory requirement is almost 10-fold lower..
- The decrease of the number of triangles in the surface mesh enables to reduce the time needed for quality rendering of the scene. Fig.9 offers diagrams that show scene rendering times with and without AM. The diagrams are built for the fixed object projection area to the mesh area ratio, which is equal to 0.1. The rendering time has been approximately reduced to a half.

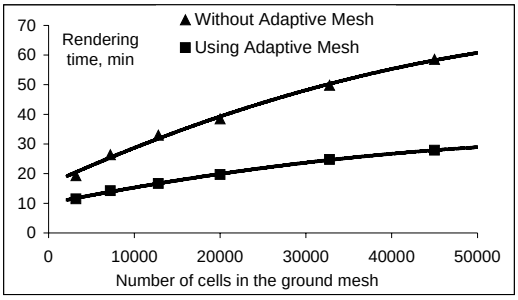


Fig.9 Scene Rendering Time With and Without AM

4.2 Real-time Visualization of Regular Terrain

The second example where the algorithm is used is a demo application for modeling the motion dynamics of a vertebrate or any other articulated vertebrate. The vertebrate is moving over terrain which is visualized with simulated flora. Quality visualization of rich dynamics of a moving vertebrate (or several vertebrates) requires complex geometry (about 1,000 textured triangles). Thus, very strict time limits imposed on the terrain modeling on the one hand and high quality requirements on the other.

The application used several algorithms to model the terrain. Initially the Finch and Bishop algorithm was used [11], [12]. After that the standard multi-triangulation algorithm for irregular data was implemented, cutting the rendering cost by a half. Finally, the algorithm described in this article was implemented (fig. 10). This algorithm decreased the terrain rendering cost even further by another 1.5 times.

The algorithm also decreased the application load time. In case of the MRM algorithm for irregular data the DAG building time was about 30 seconds and DAG size about 20MB. Regular data algorithm build DAG in 2-3 seconds.

The algorithm allows to get the desired terrain visualization quality with 300–700 triangles. However, the accuracy of the contact between the vertebrate and the terrain is not sufficient, e.g., with the coarse mesh the feet appear to penetrate inside the terrain. The notion of distance was changed to smooth out this penetration effect, i.e., the distance between the closest vertebrate to observer and the terrain point is taken as the error criterion instead of the distance between the observer view point and the terrain point. In other words, terrain discretization is performed from the point of view of the vertebrate closest to the observer. Besides, in order to decrease popping, the coarsest triangulation contains not 2 but 32 triangles.

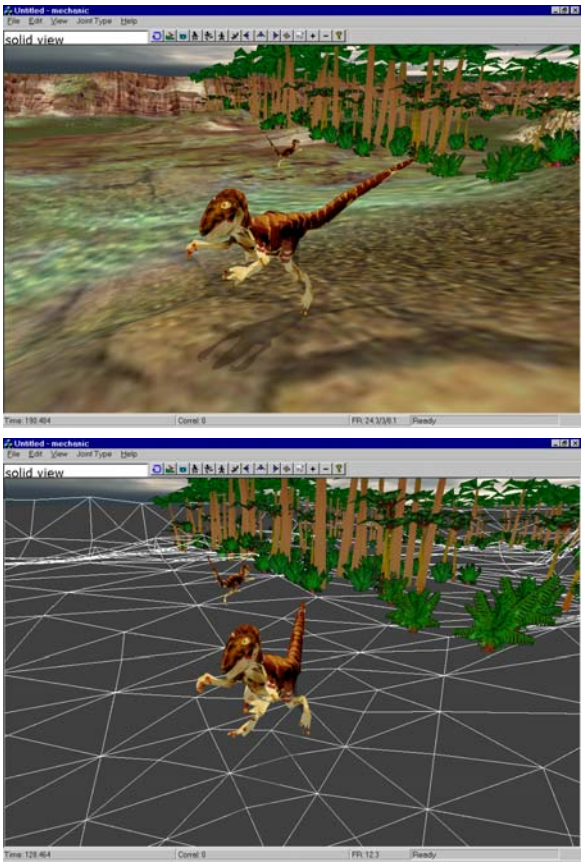


Fig.10 Example of Visualization of the Regular Terrain (algorithm suggested in this article) Above: surface with imposed texture. Below: terrain triangulation.

5. CONCLUSION

The article presents MRM implementation for regularly distributed data, provides comparisons of the performance of MRM building algorithms for regularly and randomly distributed data. Building MRM on the regularly distributed data has evident advantages since it is very time and memory cost effective. The article also gives examples to illustrate the advantages of MRM.

Our model and Adaptive Mesh in particular could be effectively used in various applications that require a considerable decrease

of the number of triangulation elements without apparent loss of quality, e.g., computer graphics, GIS, physical modeling.

6. BIBLIOGRAPHY

[1] E.Puppo, *Variable Resolution Triangulations,* 1998.

[2] A. Kapustin, J. Fedorova, T. Firsova, A. Churbanov, *An Adaptive Model for Objects representation in 3D Computer Graphics*, GraphiCon'99, Conference Proceedings, pp. 95-102, 1999.

[3] E.Puppo, R.Scopigno, *Simplification, LOD, and multiresolution – Principles and applications*, Eurographics'97 Tutorial Notes, 1997.

[4] E.Puppo, *Variable Resolution of Terrain Surfaces*, Proceedings Eight Canadian Conf. on Comp.Geom., pp.202-210, 1996.

[5] L.De Floriani, P.Magillo, E. Puppo, *Selective Refinement of Surface Meshes: Data Structures and Algorithms*, 1998.

[6] Z.T.Chen, W.R.Tobler, *Quadtree representation of digital terrain,* In Proceedings Auto Carto London, pp. 475-484, London, 1986.

[7] B.Von Herzen, A.H.Barr, *Accurate triangulation of deforming, intersecting surfaces,* Computer Graphics (Siggraph 87 Proc.), 21(4):103-110, 1987.

[8] L.De Floriani, P.Marzano, E.Puppo, *Multiresolution models for topographic surface description,* The Visual Computer, 12(7):317-345, 1996.

[9] P.Lindstrom, D.Koller, W.Ribarsky, L.F.Hodges, N.Faust, G.A.Turner, *Real-time, continuous level of detail rendering of height fields*, In Comp. Graph. Proc., Annual Conf. Series (Siggraph '96), ACM Press, pp. 109-118, New Orleans, LA, USA, Aug. 6-8, 1996.

[10] J.Fedorova, A.Kryachko, *The 3D Studio MAX plugin for simulation of the footprints on the ground surfaces*, GraphiCon'99, Conference Proceedings, pp. 296-300, 1999.

[11] M.Finch, L.Bishop, *Sorted Height Field Rendering,* 1997, http://www.ndl.com/wpapers/ndlter.html.

[12] A.Kapustin, V.Eroukhimov, A.Malashkina, *The implementation of the real-time terrain visualization algorithm*, GraphiCon'99, Conference Proceedings, pp. 110-116, 1999.

Authors:

Kapustin Alexander,

E-mail: **kap@nstl.nnov.ru**, phone# (8312) 31-90-10.

Крячко Андрей, студент Нижегородского государственного университета,

E-mail: **kray@nstl.nnov.ru**, phone# (8312) 31-90-10.

Fedorova Julia,

E-mail**: juf@nstl.nnov.ru**, phone# (8312) 31-90-10.

Firsova Tatiana,
E-mail: **tafir@nstl.nnov.ru**, phone# (8312) 31-90-10.