

Efficient Implementation of Multiresolution Triangle Strips ^{*}

Óscar Belmonte¹, Inmaculada Remolar¹, José Ribelles¹,
Miguel Chover¹, and Marcos Fernández²

¹ Departamento de Lenguajes y Sistemas Informáticos, Universitat Jaume I,
12071 Castellón, Spain

`{belfern,remolar,ribelles,chover}@lsi.uji.es`

² Departamento de Informática, Universitat de València,
46100 València, Spain

`{marcos}@robotica.uv.es`

Abstract. Triangle meshes are currently the most popular standard model to represent polygonal surfaces. Drawing these meshes as a set of independent triangles involves sending a vast amount of information to the graphic engine. It has been shown that using drawing primitives, such as triangle fans or strips, dramatically reduces the amount of information. *Multiresolution Triangle Strips* (MTS) uses the connectivity information to represent a mesh as a set of *multiresolution triangles strips*. These strips are the basis of both the storage and rendering stages. They allow the efficient management of a wide range of levels of detail. In this paper, we have taken advantage of the coherence property between two levels of detail to decrease the visualisation time. MTS has been compared against *Progressive Meshes and Multiresolution Ordered Meshes with Fans*, the only model that uses the triangle fan as an alternative to the triangle primitive. In all cases, Multiresolution Triangle Strips obtains a better frame rate.

1 Introduction

Triangle meshes have become the standard model to represent polygonal surfaces in Computer Graphics. The main reasons for this are the simplicity of the algorithms for drawing triangles, which are easily implemented in hardware, and the fact that any polygon with a number of sides greater than three can be broken down into a set of triangles.

Nowadays, highly detailed geometric models are necessary in many Computer Graphics applications. In these cases, the objects are represented by large triangle meshes, which are expensive to visualise. Some objects in the scene could be replaced by an approximation, adapting the number of triangles of a mesh to the needs of each application. This approximation is said to have a lower *Level of Detail* (LOD).

^{*} Supported by grant P1.1B2000-21 (Fundació Caixa Castelló - Bancaixa)

Multiresolution models support the representation and processing of geometric entities at different levels of detail, depending on specific application needs. The common criteria to determine the most suitable LOD are the distance of the object from the viewer, the projected area of the object on the screen, the eccentricity of this object on the screen and the intrinsic importance of the object. Current graphics systems can render more triangles than they receive. The bottleneck at the rendering stage is the throughput of the graphics systems in receiving the information to visualise. This information decreases considerably if the connectivity information between triangles is used in the mesh representation. Triangle fans and strips, which appear in the majority of graphics libraries such as *OpenGL*, make use of this property. Modelling a mesh as a set of triangle fans or strips saves having to send a large amount of redundant information to the graphics system.

All the multiresolution models in the literature, except *Multiresolution Ordered Meshes with Fan* (MOM-FAN), base the storage and rendering process on the triangle primitive. *Multiresolution Triangle Strips* (MTS) is the first model that represents a multiresolution object using the triangle strips primitive in these two stages. An MTS model is made up of a set of multiresolution strips. A multiresolution strip represents the original strip and all its LODs.

In section 2, the concept of triangle strips and the problem of searching strips in a triangle mesh are reviewed. The simplification method by Garland and Heckbert, and multiresolution models that make use of triangle fans or strips at the rendering stage are commented in this section. In section 3, MTS is presented, together with its data structure and the LOD recovery algorithm. It is shown, through the use of an example, how a multiresolution strip is constructed. In section 4, it is explained how the coherence property is incorporated into the model. In section 5, time results of MTS are shown and are compared against the results obtained for PM and for MOM-FAN. Finally, in section 6, conclusions and future work are presented.

2 Previous Work

In this section, some works relative to the multiresolution model presented in this paper are reviewed. The concept of triangle strips and an algorithm for searching strips over a polygonal surface is discussed. After that, we review the simplification algorithm of Garland and Heckbert [4], which has been used to obtain the coarse meshes from the original model. Finally, other multiresolution models that use triangle fans or strips, in either the storage or the rendering stage, are discussed.

2.1 Triangle strips

A strip encodes a sequence of triangles where every two sequential triangles share a common inner edge. Making use of this connectivity information decreases the amount of information sent to the graphic engine at the rendering stage. A



Fig. 1. Left, sequential strip. Right, generalized strip

triangle strip is encoded by a sequence of vertices. The vertex sequence for the strip in Figure 1.a is 0,1,2,3,4,5, where triangle i is made up of the vertices i , $i+1$ and $i+2$. In this way, it is only necessary to send $T+2$ vertices to the graphic engine in order to render T triangles, as opposed to the $3T$ vertices required if the triangles are rendered independently. The sequence of vertices for the strip in Figure 1.b is 0,1,2,3,4,3,5,6. In this case, it is necessary to send vertex 3 twice. This operation is called *swap*. A triangle strip that can be represented without any swap operation is called a *sequential strip*. A triangle strip that needs this operation is called a *generalized strip*.

2.2 Searching Triangle Strips

A triangle mesh can be drawn as a set of triangle strips. The best set of triangle strips that describes a mesh sends the lowest number of vertices to the graphics system. Searching this best set of strips in a mesh is an NP-complete problem [1]. Thus, it becomes necessary to use heuristic strategies in the search for strips in a rational time. The STRIPE algorithm [3] has been used for extraction of the triangle strips. This algorithm is publicly available at <http://www.cs.sunysb.edu/~stripe/>. The STRIPE algorithm allows control over some parameters in the searching process, such as the use of swap operations.

2.3 Simplification Using Vertex Pair Contraction

A simplification method provides a version of the original object with fewer triangles while maintaining the visual appearance. Important surveys about simplification methods [4] [5] [9], have been published. The simplification method used in Multiresolution Triangle Strips is the method proposed by Garland and Heckbert [4] based on vertex pair contraction. This method, called *Qslim*, has been used for several reasons: i) it returns the sequence of contractions applied to the original object, ii) the simplified mesh error is sufficiently small, and iii) it is public domain and can be found at <http://www.cs.cmu.edu/~garland/quadrics/>.

2.4 Multiresolution Modelling

Garland [6] defines a multiresolution model as a model representation that captures a wide range of approximations of an object and which can be used to reconstruct any one of them on demand. Multiresolution models can be classified

in two broad groups: *discrete multiresolution models*, which contain a discrete number of LODs and a control mechanism to determine which LOD is the most adequate in each moment; and *continuous multiresolution models*, which capture a vast range of, virtually continuous, approximations of an object. These can be subdivided into two main classes according to their structure: tree-like models, and historical models [9]. There is no relationship between the LODs of the object in a discrete multiresolution model. Thus, the size of these models increases quickly when new levels of detail are included. They usually store between five and ten levels of detail. Graphics standards such as *VRML* or *OpenInventor* use discrete multiresolution models. These models are easily implemented and can be edited by users and optimised for rendering. The main disadvantage is the *visual artefact* that occurs during the change between two levels of detail. One solution to decrease this visual artefact is to draw both levels of detail of the object using transparency methods.

Two consecutive levels of detail differ by a lower number of triangles in a continuous multiresolution model. These little changes introduce a minimal visual artefact. The size of the model decreases as compared to the discrete models because no duplicate information is stored. The Progressive Meshes of Hugues Hoppe [7] is the most well known continuous multiresolution model available nowadays. It is included in Microsoft's graphics library DirectX 8.0.

Multiresolution Models Using Triangle Fans or Strips Primitives. Hoppe [8] has used strips in the rendering stage of a viewdependent multiresolution model. After selecting which triangles to draw, strips of triangles are searched. Through experimentation, Hoppe concludes that the fastest triangle strip search algorithm is a greedy one. In order to reduce the strip fragmentation, strips are grown in a clockwise spiral manner.

El-Sana et al. [3] have developed a view-dependent multiresolution model based on an edge-collapsing criterion. The first step in constructing the model is to search triangle strips on it. These triangle strips are stored in a data structure called a skip list. Once the multiresolution model has determined which triangles to visualise, the skip list is processed. If none of its triangles has been collapsed, the strip is drawn and if not, the skip list is processed in order to update the strips. The triangle strips are not the basic primitive of the model, they are only used to speed up the rendering process. The work presented in [10] uses triangle fans as its basic representation primitive. Using this primitive, the storage cost is reduced, but the behaviour of this new model as regards its visualisation time is similar to its predecessor. A short average fan length, the high percentage of degenerate triangles, and the need to adjust the fans to the required LOD in real time all contribute to produce overall results which do not bring about a global improvement in visualisation time.

Neither of the previous models uses the triangle strips primitive in both the storage and the rendering stage. Hoppe searches the strips over the simplified model prior to rendering it. In El-Sana's work, the triangles Skip-Strips are updated for the collapsed edges before rendering the model.

3 MTS Model

An MTS model represents a mesh as a set of multiresolution triangle strips. Each multiresolution triangle strip is made up of the original strip and all its LODs, and a list containing the vertices at the beginning of the strip for each LOD. The sequence of vertices in a strip induces an order relationship between them. It can be concluded that these strips may be represented by a directed graph. Due to this fundamental structure, the LOD recovery algorithm is based on graph traversal.

3.1 Data Structure

A multiresolution strip is represented by a directed graph and a list with the vertices at the beginning of the strip.

Conceptually, each vertex of the strip is identified by a node on the graph and each inner edge with an arc on the graph. Two nodes are said to be *adjacent* if they are joined by an arc. If it is a directed graph, the arc direction is determined by the order of the vertices in the sequence that represents the strip. In practice, a graph is represented by an adjacency list [2], (see Fig. 2.c). In this representation, each node v has a list with all of its adjacent nodes w_i for any LOD. Each node w_i on the adjacency list of node v represents an arc from v to w_i . Each strip in the model is an instance of the *class MultiresolutionStrip*.

Each node on the graph has three fields. The first field (*vIndex*) is an index to the memory address containing the geometric data of the vertex. The second field (*neighbours*) is a pointer to the adjacency list. The third field (*currentNeighbour*) is an index to the next neighbour that will be visited by the LOD extraction algorithm. Each node on the graph is an instance of *class ColumnNode*, (see Listing 1).

```

class ColumnNode
    unsigned long vIndex;
    RowNode * neighbours;
    unsigned int currentNeighbour;
end class

class MultiresolutionStrip
    RowNode * sBegin;
    ColumnNode * colVertices;
end class

class RowNode
    unsigned int colIndex;
    unsigned long res;
end class

```

Listing 1: Data structures of the MTS model.

Each arc on the graph has two fields. The first field (*colIndex*) is an index to the next node in the vertex sequence that represents the strip. The second field (*res*) is an integer that indicates the maximum LOD at which the arc may be traversed by the LOD extraction algorithm. Each arc of the graph is an instance of *class RowNode*.

The vertex at the beginning of the strip could change as the strip is simplified. The list of strip beginnings indicates, for each LOD, which is the vertex at the beginning of the strip. Each element on the list of strip beginnings has two fields. The first field (*colIndex*) is an index to the node that begins the strip. The second field (*res*) is an integer indicating the maximum LOD at which the vertex is valid as the beginning of the strip. These fields are the same that those in the *class RowNode*, so each element on the list of strip beginnings is an instance of *class RowNode*.

Construction Example. The construction of a multiresolution strip starts from the original strip. As the strip is simplified, the sequence of vertices changes. These changes in the vertex sequence should be introduced into the graph and the list of strip beginnings that represents the strip.

Let's take the strip in Fig. 2.a) as an example. In this figure, the initial graph and the strip beginning list are shown. Let's label the maximum level of detail at which a strip can be represented as level of detail 0 (LOD 0). The sequence of vertices at LOD 0 is 0,1,2,3,4,5,6,-3. The special label -3 specifies that the end of the strip has been reached. After the first vertex pair contraction ($2 \rightarrow 3$), LOD 1, the resulting vertex sequence is 0,1,3,-1,3,4,5,6,-3. The special node labelled -1 indicates that vertex 3 is the end of one strip and the beginning of the next one. There are two new edges (1,3) and (3,-1,3,4). The new edge (1,3) is stored in the multiresolution strip by adding a new arc to the list of adjacencies of node 1. The field *colIndex* of this new arc is initialised to 3 and the *res* field is initialised to 1. The special arc (3,-1,3,4) is stored as follows: first, a new arc is added to the list of adjacencies of node 3. This arc has its *colIndex* field initialised to -1 and its *res* field initialised to 1. Another new arc is also added to node 3, in this case with its *colIndex* field initialised to 4 and its *res* field initialised to 1. The arcs that remain, (4,5) and (5,6), update their *res* field to the new LOD 1.

The second vertex pair contraction ($0 \rightarrow 3$), LOD 2, produces the vertex sequence 3,4,5,6,-3. The beginning of the strip has become vertex 3. This change is stored in the multiresolution triangle strip and a new object of *class RowNode* with its *colIndex* field initialised to 3 and its *res* field initialised to 2 is added to the list of strip beginnings. Three edges disappear while the others remain. This fact is stored in the graph by updating the *res* field of the remaining arcs to the new LOD 2.

A new strip having only one triangle is produced after the last contraction ($6 \rightarrow 4$), LOD 3. The vertex sequence is 3,4,5,-3. The arcs (3,4) and (4,5) remain at this LOD so its *res* field is updated to 3. The final vertex of the strip at this LOD is vertex 5, and a new node with the special label -3 has to be added to the list of adjacencies of this vertex.

3.2 Uniform Level of Detail Recovery Algorithm

The LOD recovery algorithm traverses the graph in order to extract the demanded LOD. The algorithm proceeds in two steps. First, the algorithm finds

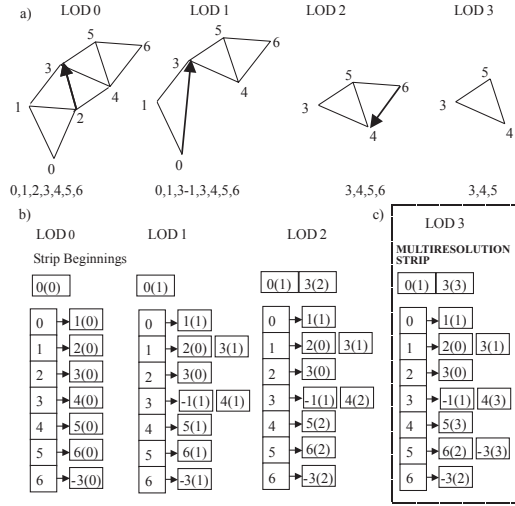


Fig. 2. Construction example of a multiresolution strip. a) Original strip and the sequence of vertex pair contractions. b) The detailed process of a multiresolution strip construction. c) The final multiresolution strip.

the vertex at the beginning of the strip that is compatible with the demanded LOD. Second, the graph is traversed from the vertex at the beginning through those arcs that are compatible with the demanded LOD until a special -3 node is reached. In this case, being compatible means that the arc in the adjacency list has a res field that is bigger or equal to the demanded LOD. The pseudo code of the algorithm is shown in Listing 2.

4 Coherence

Coherence in a multiresolution model means taking advantage of the last information extracted in the extraction of the new LOD demanded. The use of coherence decreases the time consumed by the recovery algorithm, thus saving the need to repeat calculations already done. Coherence in an MTS model means only processing those multiresolution strips that have changed with respect to the last LOD extraction. The sequences for those multiresolution strips that remain will be the same. Each sequence of a multiresolution strip has an LOD interval of validity. The minimum of the interval (minRes) is the maximum value of those arcs that are not compatible with the demanded LOD. The maximum of the interval (maxRes) is the minimum value of those arcs that are compatible with the demanded LOD. In this way, if the new LOD demanded is between these two values the vertex sequence remains valid. The minimum and maximum of the interval are retrieved when the graph is traversed. This interval is recalculated every time it becomes invalid.

```

{ First we search the strip beginning }
while beginning.res < res and Beginnings  $\neq \emptyset$ 
    beginning = nextbeginning;
end while
if beginning Not Found then { This strip does not }
    exit
else { exist at this resolution }
    Node = beginnning;
    while not EndStrip { While there are vertices in strip }
        if Node is not special node then
            DrawVertex;
        else if Node is -1 or Node is -2 then
            NewStrip;
        else if Node is -3 then
            EndStrip = true;
        end if
        Neighbour = Node.currentNeighbour;
        while Neighbour.res  $\geq$  ResolutionDemanded
            Neighbour = nextNeighbour;
        end while
        Node = Neighbour;
    end while
end if

```

Listing 2: Pseudo-code of the level of detail recovery algorithm.

5 Results

The MTS model has been subjected to several tests. These tests are addressed at evaluating the visualisation time in a real time visualisation application. The results are compared with those of the PM and MOM-FAN. PM uses the triangle primitive both in the data structures and at the rendering stage. MOM-FAN uses the triangle fans primitive both in the data structures and at the rendering stage. A proper implementation of PM is used. Previously, this implementation has been verified. The time obtained in the extraction is the same as that published by the author. The polygonal objects used in the test come from the *Stanford University Computer Graphics Laboratory* (<http://www-graphics.stanford.edu/data/3Dscanrep/>) and *Cyberware* (<http://www.ciberware.com/models/>). The tests were performed on an HP Kayak XU with two Pentium III processors at 550 MHz and 1 Gb. of main memory. A GALAXY by Evans & Sutherland with 15 Mb. of video memory was used.

5.1 Time Extraction Reduction Using Coherence

Table 2 shows the extraction time with and without coherence for the MTS models. This time remains almost constant independently of the number of LODs extracted for models without coherence. This is due to the fact that every mul-

Table 1. Characteristics of the objects and model sizes in Mb.

Model	#Strips	#Triangles	#Vertices	MTS	PM	MOM-FAN
Cow	136	5804	2904	0.263	0.256	0.202
Sphere	173	30624	15314	1.301	1.318	1.104
Bunny	1229	69451	34834	3.095	2.993	2.245
Phone	1747	165963	83044	16.995	16.770	13.309

Table 2. Time extraction reduction using coherence in the MTS model. Time in milliseconds.

Model	Linear		Exponential		Random	
	Coherence	Without	Coherence	Without	Coherence	Without
Cow	0.469	1.519	0.389	1.713	1.560	1.640
Sphere	7.364	10.209	6.234	10.830	10.700	10.080
Bunny	15.197	27.120	13.052	29.067	27.500	27.965
Phone	44.264	65.539	38.613	70.254	65.470	66.410

tiresholution strip is processed for each new LOD demanded. If coherence is used, only multiresolution strips that have changed are processed.

5.2 Visualisation time

The MTS model has been compared against PM and MOM-FAN models. These models have been subjected to two kinds of experiments. The first experiment is addressed at evaluating the performance of each model in terms of number of frames per second when a series of LODs are demanded and the object is immediately visualised. The second experiment is more realistic. The user of a real time application needs to feel smooth, not jerky, motion. In this experiment a new LOD is demanded each second. After the extraction, the model is visualised until a new LOD is demanded. Results are shown in Figure 3.

6 Conclusions and Future Work

The main contribution of the multiresolution model presented in this paper is the use of the triangle strips primitive as the basis of the data structure and in the rendering stage. The main benefit of using the triangle strip primitive is the decrease in the amount of data sent to the graphic engine and the resulting acceleration of the rendering stage.

The MTS model has been compared with PM and with MOM-FAN. The model sizes created with MTS are comparable with those of PM. As regards visualisation time, MTS provides significantly higher frame rates than those offered by PM. MTS model sizes are bigger than those created by MOM-FAN. Therefore, MTS frame rates are also significantly higher than the results obtained

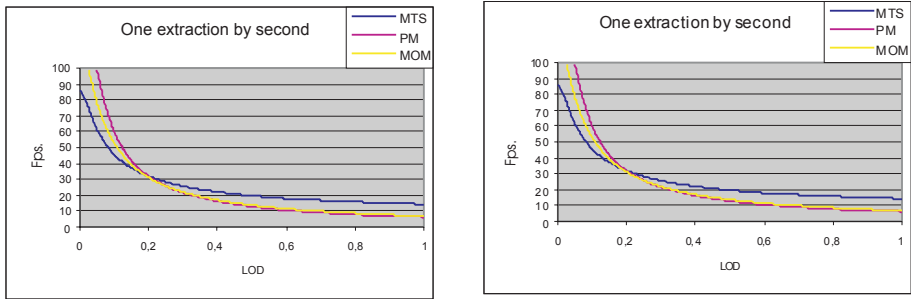


Fig. 3. The plot on the left shows the frame rate for each model as a new LOD is demanded and visualised. The plot on the right shows the frame rate for each model when an LOD is demanded each second. One hundred LODs were extracted.

by MOM-FAN. This is mainly due to the fact that triangle fans in an MOM-FAN model have between three and four triangles per triangle fan. The number of triangles per strip in MTS is about forty. The main conclusion is that the use of triangle strips in a multiresolution model provides frame rates far beyond those offered by models based on the triangle or triangle fans primitive. The drawback of the MTS model is the high extraction time. Introducing coherence inside each strip is something to be dealt with in future work in order to reduce the extraction time. View-dependent visualisation and progressive transmission of the models through a computer network are also lines of future work.

References

1. Arkin E.M., Helod, M., Mitchell J. B. S., Skiena, S. S.: Hamiltonian Triangulation for Fast Rendering, *Visual Computer* 12(9), 429-444, 1996.
2. Brassard, G., Bratley P.: *Fundamentals of Algorithmics*, Prentice Hall, 1996.
3. El-Sana, J. Evans, F., Varshney, A., Skiena S., Azanli, E.: Efficiently Computing and Updating Triangle Strips for Real-Time Rendering. *The Journal Computer-Aided Design*, Vol(32), IS(13), 753-772.
4. Garland, M., Heckbert P.: Surface Simplification Using Quadratic Error Metrics. *Proc. of SIGGRAPH'97 (1997)* 209-216
5. Garland, M., Heckbert, P.: Survey of polygonal surface simplification algorithms, *Multiresolution Surface Modeling Course Notes of SIGGRAPH'97*, 1997.
6. Garland, M.: Multiresolution Modeling: Survey & Future Opportunities. *State of the Art Reports of EUROGRAPHICS '99 (1999)* 111-131
7. Hoppe, H.: Progressive Meshes, *Proceedings of SIGGRAPH '96*, 99-108, 1996.
8. Hoppe, H.: View-Dependent Refinement of Progressive Meshes. *Proc. of SIGGRAPH'97 (1997)* 189-198
9. Puppo, E., Scopigno, R.: Simplification, LOD and Multiresolution - Principles and Applications, *Tutorial Notes of EUROGRAPHICS'99*, 1999.
10. Ribelles, J., López, A., Remolar, I., Belmonte, Ó., Chover M.: Multiresolution Modelling of Polygonal Surface Meshes Using Triangle Fans, *Proceedings of 9th Discrete Geometry for Computer Imagery Conference*, 431-442, 2000.