Exploring the use of Reeb graphs in LOD generation

Benoit Lange, Nancy Rodriguez

HAL Id: lirmm-00450487

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00450487v1

Submitted on 26 Jan 2010

Exploring the use of Reeb graphs in LOD generation

Benoit LANGE
LIRMM
LIRMM, UMR 5506 - CC 477
161 rue Ada
34392 Montpellier Cedex 5 - France

Benoit.lange@lirmm.fr

Nancy RODRIGUEZ
LIRMM
LIRMM, UMR 5506 - CC 477
161 rue Ada
34392 Montpellier Cedex 5 - France

Nancy.Rodriguez@lirmm.fr

ABSTRACT

Level of detail was one of the fundamental solutions of simplification 3D scene. But some issues with this solution are present. For example the topology of the mesh is not kept. In this paper we introduce the Reeb graph in level of detail. Reeb graph is a topological extractor for graph. We extract the most important point of a mesh and use them as LOD. These points are representing by a skeleton call Reeb Graph. We make a related work on LOD, skeleton extraction and extraction of Reeb graph. After, we introduce our work. We define augmented LOD or LOD+. We present the results of the different solutions that we implement like: LOD 0, LOD +, and measure between LOD. LOD 0 is the lowest LOD in a scene. LOD + consist by applying a skeleton on a 3D mesh at low resolution. And measure different between two match allow to compare different solutions of simplification.

Categories and Subject Descriptors

I.2.10 [**Vision and Scene Understanding**]: Extraction of Reeb graph in 3D model

Keywords

Level of detail, LOD, Reeb graph, skeleton, Morse theory, OFF, QEM, Garland.

1. INTRODUCTION

The modern graphics hardware is able to display thousands of polygons but they are limited in number. It is needed to simplify the 3D scene. Level of detail (LOD) is one of the most important features in graphics computation. This method is used for render large 3D model. In 1976, James Clark introduces the LOD concept [1]. It was use for produce different resolution for model. This is done by reducing the numbers of vertices and edge, this solution allow to understand easily shape of the mesh.

The most common use in LOD is global or local simplification. This kind of simplification is done on the mesh vertices or edge. But, in this paper, we use skeleton for LOD generation. Skeletons were mostly use in animation, classification, shape analysis, compression or collision detection. Skeleton can be compute or define by designer. Here, we will compute it using Reeb Graph extraction. It was introduce in 1946 by George REEB. [2] explain how to extract

topology of a graph. It was an extension of Morse theory. But we can use this extractor for 3D model, it allow computing the most important point of the mesh.

This paper is decomposed in three parts.

- Related work: we introduce LOD, extraction of skeleton and Reeb graph.
- Reeb graph in LOD: we explain our four uses of Reeb graph in LOD filed.
- Results and discussion.

2. Related work

Many algorithms have been produce for solving the low possibility to render large meshes. David Luebke makes some survey on LOD solutions [3, 4]. Filter shape is the solution to simplify meshes. It's exist three main filtering solutions.

The first solution is space division. It was use with octree, quadtree. It's a space division of the mesh in a tree data structure. This kind of structure allow to define witch object is seen by the camera. It's allowed to compute easily and quickly object in the scene.

Another solution is object visibility: painter algorithm or Z buffer. This two techniques use occlusion of object for hide none visible face.

The last solution study was Level of detail (LOD). It was introduced by James Clark in 1976 [1]. This solution uses many different resolutions of 3D in the scene. It necessary to define some distance between the camera and the mesh and after the algorithm display the right low resolution mesh.

This part will be divided in three parts.

- Level of detail
- Skeleton extraction
- Skeleton extraction with Reeb graph

2.1 Level of detail

More and more project produce better 3D models like: the digital Michelangelo project that produce a mesh with 2 billion polygons. Actual graphics card aren't able to display this kind of mesh. And it's needed to simplify them. Figure 1 show different model resolution according to the distance.
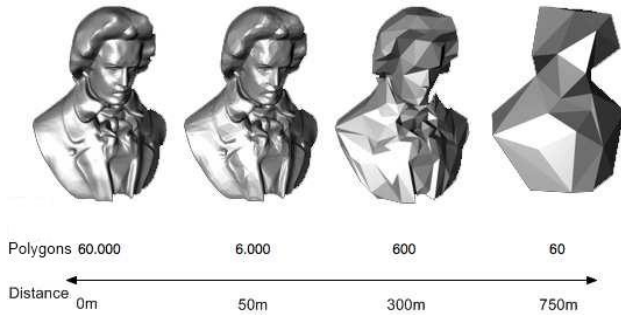

Figure 1. Different resolution of a mesh

LOD was often release by the designer but some algorithms can produce them with keeping some important features. It exist two kinds of LOD: continues and discrete.

Discrete LOD produce the simplified model in preprocess. It was the most use solution due to the computation cost. This kind of algorithm is not in real time. We choose the number of vertices and edge and the algorithms produce models. The main issue with these algorithms is the popping effect. This effect is a visual issue: the difference between two resolution mesh is visible if the number of face is too different. On Google earth, we can see it when we look 3D buildings. Some solutions exist to reduce this effect. We call this morphing. It consist by add vertices in real time when the resolution of the mesh change.

The other solution is continuous LOD. This method was firstly defined by Luebke. It uses a view dependent simplification to produce a mesh. This solution simplifies the entire scene instead of a individual model. These algorithms are in real time and solve the problems of popping. But the computation is expensive. And simplification in real time is hard to be done.

Then in LOD it exist two types of simplification local and global.

Local simplification is use in [5-10]. The algorithms remove, collapse, split or swap edge. These simplifications are done locally on the mesh. This can produce some hole …

[5] is one of the first algorithm to simplify mesh. He produces multiple removing of vertex on the mesh. He done is job in three steps. Each vertex can be removing but the algorithms give some width to each one in function of their importance: complex, simple, boundary … The limitation of this simplification is the poor fidelity of the mesh.
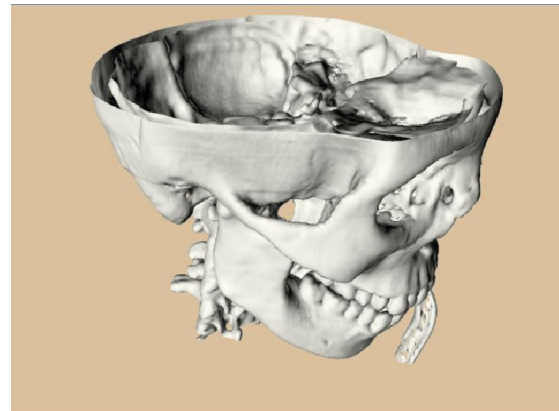

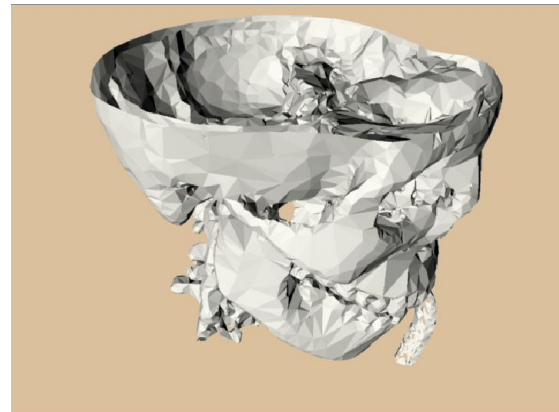**Figure 2. Original Mesh**

Figure 3. Mesh with 90% decimation

[6] is a decimation solution with a sampling method. The algorithms put on the mesh vertex, who the user defines, spring are applied on each new vertex by following the original mesh. And the method mutual tessellation produces the final model.

Many other algorithms exist and preserve topology of the mesh or some features like curves …

Another way to simplify mesh is global simplification. This approach is more recent. It allows

simplifying a global mesh. For example simplifications are done on the entire surface and not locally like the previous one. We can find this technique in [11-16]. Many solutions exist to simplify the global mesh: with hierarchical model, by voxel filter, by envelope simplification or base mesh.

[14] is one of this solution. It consists by reducing the number of vertices by producing two envelopes:

- Inside the mesh
- Outside the mesh

They are used for simplify model. The algorithm applies a simplification on the mesh. If the vertex goes out an envelope, he was bringing back between them. The big problem with this solution is the computational cost and the difficulty to extract the two envelopes.

Figure 4. Simplification envelope

[16] is introduce by Hoppe. Hoppe is one of the famous Lod algorithm producers. Many of this solution were use and upgrades. Here the method produce a base meshes and adds vertices incrementally. This can produce a better resolution for a mesh and keep the main important point of the mesh.

After this related work on LOD, we will talk about Skeleton extraction. This method is used for many problems.

2.2 Skeleton extraction
Skeleton extractions have different use graphical computation. They are use to work on the mesh with make hard computation due to the numbers of vertex.

In this part we will present the usage of skeleton extraction, some algorithms to produce this object and at last the Reeb extraction.

2.2.1 Usage of Skeleton extraction:
For extracting bones of a mesh, two ways are use: designer produces it by their hand or an algorithm can compute it. After the usage of this kind of structure we will present method to produce bones of the mesh.

It exist at least five usage of skeleton extraction:

- Mesh classifier
- Mesh animation
- Shape analysis
- Compression mesh
- Collision detection

The first one was classification. Here we use skeleton to recognize propriety of an object. Some algorithms produce a mesh classifier and can identify some 3D model without any human action. It allow to index mesh libraries.

A second way to use bones extraction is animation. This technique is use like bones in human body. Some part of the mesh where assign to a bone and they move like it. This solution is often create by designer because they draw there mesh around a skeleton. But some models don't own their movement axis, and it was needed to extract them.

Mesh compression is another solution of skeleton extraction. This solution is use for transfer mesh in a little file. The skeleton was the non compressible data of the mesh and each vertex are encoded in the tree.

Extracting skeleton can help for collision. Often, we use hull, box or sphere to detect the collision. Algorithm like Qhull use voronoï diagram to produce the hull. The problem with this solution is the non realistic collision issue. An animal cannot be represent by a box due to is tree structure. So use a skeleton is more realistic collision detection. The width of the bone is a problem due to the thickness of it.

Finally, the last use is shape analyze. This method allows extracting some important feature on the mesh: we can identify the curve or something else. The main use for extract them is REEB algorithm. In this paper we will talk about this solution.

In the next part, we will talk about different way to extract skeleton.

2.2.2 Skeleton extraction:

Many solutions exist for extract the mesh skeleton. Here we will talk about two methods.

[17] is middle axe geodesic extraction. He produces the middle axis of each geodesic surface of the mesh. After this we compute the middle axis by align each geodesic axe. This method is one of the first who the mesh position is not important. With this solution we can identify some shape of the mesh like hand.

[18] is another solution who compute the middle axis of the mesh by using spring on each vertex. After a few iterations, this algorithm aligns bones and aligns important axis like shoulders or something else. This kind of algorithms is mostly use for animals.

Other solution for extracting skeleton is topological extraction. For doing this we use [1] and we extract le main important points. This graph is call Reeb graph. He was a evolution of Morse theory on manifold graph. This solution extract singular point of a shape .

Morse theory was introduced in 1925 by Marton Morse. This theory infers manifold's topological invariant from a Morse function. In [19], Tierny give an mathematical translation of the Reeb graph theory.

In fact Reeb graph are compose from nodes that are the critical point of the structures. And they are the nodes of the reeb graph.

Critical points are extracting from a special function like diameter of a graph. For each point of the long way of the graph, we compute critical points by using gradient function.

It exist three critical points to identify. The first one was bijection point. It's a point who the next step, on the diameter way, separate himself to produce a branch.

Another important point is merging point. The branch of the tree becomes a single point.

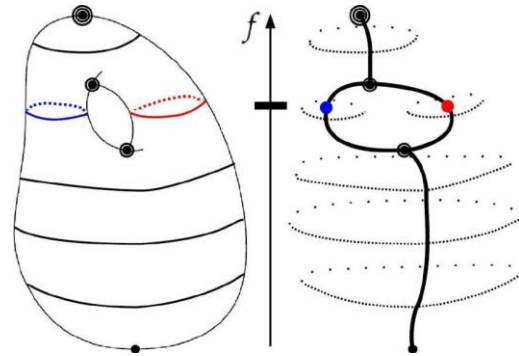And finally the next step try to identify the end point of the mesh.

Figure 5. Reeb graph of a simple shape

On this picture we can see the three kind of points and the graph produce by the algorithm.

The characteristic points are compute from the two point of the diameter by usage of gradient function.
In this paper, we will talk about some few implementation of the Reeb graph.

Two mains solutions exist:

- High level Reeb graph [19,20]
- Harmonic Reeb extraction[21]

Julien Tierny work on Reeb extraction during is PHD. He use is own function method to compute the Reeb graph. The results of this extractor are really good.

The other solution who have find is Harmonic skeleton extraction. This solution is defined for animals. It was the only one solution available in open source.

3. Reeb graph in LOD:

In this paper we will introduce four new methods who use Reeb Graph: LOD 0, LOD +, Drive LOD and measure of LOD.

We use harmonic extraction of Reeb graph but these techniques have some problems.

In fact mesh need to be manifold and it's not easy to find this kind of models.

Another issue was file format. The simplification use for simplify is QEM from Garland who use SMF files and the harmonic extraction of skeleton use OFF files. We need to use a file convertor to produce correct files.

Skeleton have to type define, so we need to create a special file format for it. We define a format like OFF file call Skel file.

SKEL

V 0 0 0 0
V 1 1 1 1
V 2 0 1 1
E 0 1
E 1 2

The V lines are for vertices and the E line are for edges.

Now, we will talk about the first part of our contribution: LOD 0. All of our new usage of Reeb graph uses several meshes: a hand, a boy, a chair, a duck, a horse and a CAD object. Before compute our algorithms, we can identify 2 kind of objects: with many important topological point (ex: hand, boy, chair or horse) and the other one (ex: duck, CAD). The first class can be assimilating like a tree.

3.1 LOD 0:

Level of detail has a big issue. If the algorithms make too hard simplification the shape wasn't keep, and the topology of the mesh become totally wrong. The model become impossible to understand and is usage is totally useless. Our Solution consists by using a very simple mesh for view the model at a far distance. This method was call LOD 0 or LOD max.
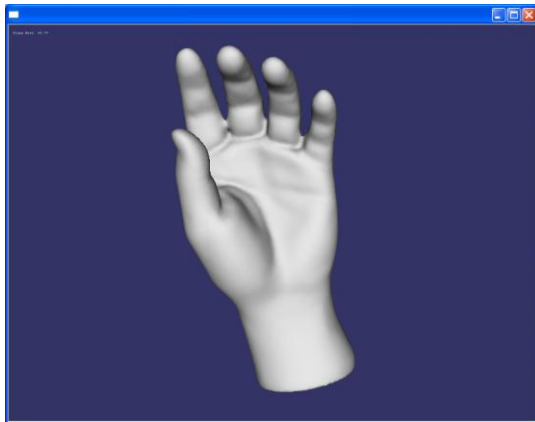

Figure 6. Hand in full resolution

For doing this, we use Reeb graph as the lower LOD of a mesh. Often billboard are use but it's not really efficient. For this solution two steps are needed:

- Extracting Reeb graph.
- Define a display distance.

Figure 6 is the original mesh and figure 7 present is Reeb graph in the scene.
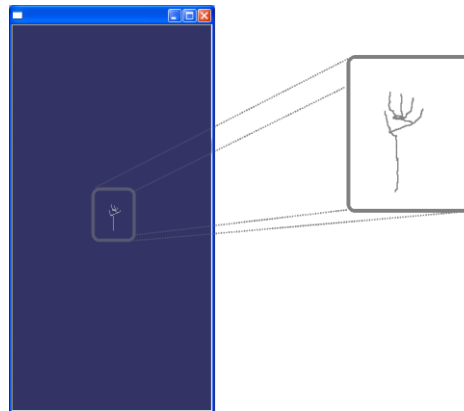

Figure 7. Hand Reeb graph at far distance

The first are good on visualization but the width of the hand is not conserved. Two problems are popping from this solution.

The first one was width. Figure 7 show that the width of the edge is important. So we decide to use some tricks to dress the tree. We use three "clothes": box, cylinder-ball, and cylinder. Cylinder and balls produce good result in shape recognition but are cost expensive. Other solutions are not efficient: they don't respect the shape of the mesh and produce some artifact like holes.

The second problem was on the kind of mesh. For memory we define two mesh classes: tree and none tree. The trees meshes have really good results, we can find 5 fingers on the mesh while QEM merge them. But with none tree meshes, we cannot imagine the shape. Figure 8 and 9 show the CAD piece and the Reeb graph of it.
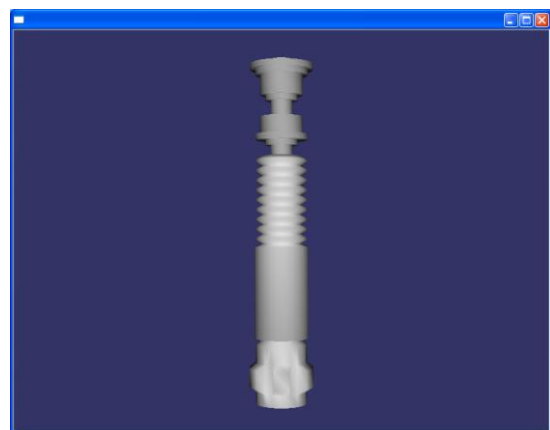

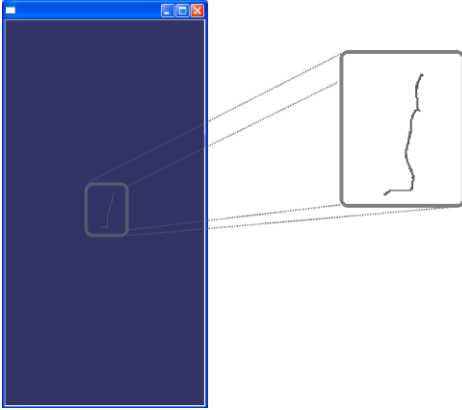**Figure 8. CAD piece at full resolution**

Figure 9. Reeb graph of CAD piece

The skeleton of the CAD piece is representative of the mesh but without "clothes" the shape of the mesh is not identifiable.

In the next part we will talk about the next solution of our solution.

3.2 LOD +:

A new feature in LOD domain is Augmented LOD or LOD +. He was the same solution like augmented reality but we set the skeleton on the simplified mesh. This solution allows keeping some important features: for example when we compute the hand with QEM, fingers disappear or merge.
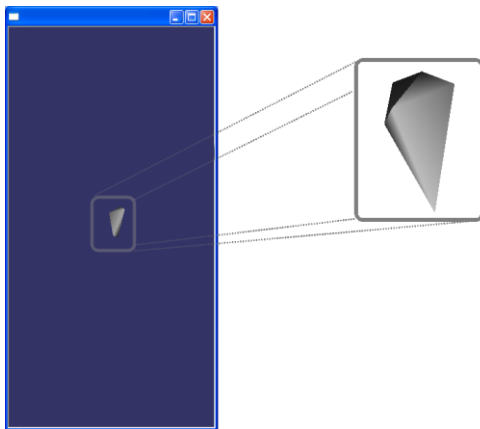

Figure 10. Hand at low resolution

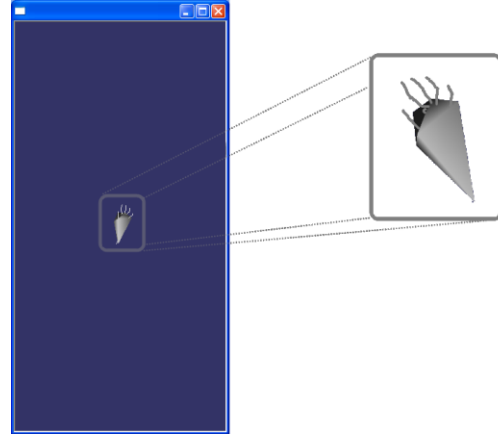Figure 11 present the LOD + who give back to the mesh the fingers.

Figure 11. LOD +

With this solution we don't lost anymore the important stuff of the mesh. But two problems income:

- The first was due to the width of the mesh like the first solution.
- The second issue was on none tree meshes. We don't find the important features of the mesh and this technique was not useful.

The next step will introduce quickly a solution that we haven't implement.

3.3 Assisted LOD:

This solution consist to drive algorithm in chooses of simplification.

Many algorithms use their own solution for drive the simplification. We imagine some solution on many algorithms.

For like [11]: the clustering grid can be compute by using Reeb graph. This solution allows the simplification to be done on the important features of the mesh. Topology of the mesh was conserved and the simplified mesh will have a good visual aspect.

In [9], we can compute the graph each time that we remove an edge. This allows measuring the topological modification of the simplified mesh at each modification. One of the main issues of this solution is the computation cost. Reeb graph is $O(N(\log(N))$, if we apply the same algorithms at each topology modification, the cost will be too expensive. Some heuristics can be used to not recompute the entire graph.

But many other algorithms, who didn't talk, can use Reeb graph. The last exploration was measure between two mesh.

3.4 Measure of a mesh and his simplification:

In this part, we will talk about a new solution to measure different between an original mesh and is simplification. Two previous solutions exist.

The existent solution use simplification method or sampling techniques. Here we want to compute the Reeb graph of a mesh and is simplification. And we compare the two graphs. More graphs are similar more the simplification have been efficient.

The first step of algorithm is to compute the two skeletons. After, we try to approximate a theoretical Reeb graph with some parameters: numbers of vertices, location of vertices. We compute the new graph and compare to the simplified tree.

With this solution we can identify witch cluster contain the most problems. The other solutions of measure don't identify the branches that have troubles.

Other ways to identify problems are possible but we don't study them in this paper.

4. Results and discussion

We introduce four approaches to use Reeb graph in LOD. We have implemented three of them. And results are quite good.

We find that some meshes aren't usable with some techniques: none tree mesh. But other models have many benefits of our solutions.

For LOD 0, the solution is good for tree meshes but its need to find a way to wear the skeleton because it's too thin for show at far distance.

For LOD +, we can restore some important part of the meshes. Little details that disappear with simplification method are keeping.

And finally, measure LOD is efficient. She allows matching the branch with the biggest issues. Our technique is not the best but she gives some interesting results.

On our future work, we want to study new implementation of Reeb graph [20].

We want to implement the assisted LOD witch many algorithms like: [11,12,13]. This allows to computer better low resolution mesh instead of using their metrics. The topological cost can be the key of mesh simplification to produce better LOD models.

Another solution is to implement other solution to compute metrics between a object and the simplified mesh.

6. REFERENCES

[1] James H. Clark. Hierarchical geometric models for visible surface algorithms. Commun. ACM, 19(10) :547_554, 1976.

[2] Reeb G.: Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. Comptesrendus des Séances de l'Académie des Sciences 222 (1946), 847–849. (Cited pages 54, 76, 78, 164, and 165.)

[3] David Luebke. A survey of polygonal simpli_cation algorithms. Technical report, Chapel Hill, NC, USA, 1997.

[4] David P. Luebke. A developer's survey of polygonal simpli_cation algorithms. IEEE Computer Graphics and Applications, 21(3) :24_35, /2001.

[5] William J. Schroeder, Jonathan A. Zarge, and William E. Lorensen.Decimation of triangle meshes. In SIGGRAPH '92: Proceedings ofthe 19th annual conference on Computer graphics and interactive techniques, pages 65_70, New York, NY, USA, 1992. ACM.

[6] Greg Turk. Re-tiling polygonal surfaces. SIGGRAPH Comput. Graph., 26(2) :55_64, 1992.

[7] Paul Borrel jarek R. Rossignac. Multi-resolution 3d approximation for rendering complex scenes. Springer Verlag, pages 455_465, 1993.

[8] Kok-Lim Low and Tiow-Seng Tan. Model simpli_cation using vertex-clustering. In SI3D '97 : Proceedings of the 1997 symposium on Interactive 3D graphics, pages 75__., New York, NY, USA, 1997. ACM.

[9] Michael Garland and Paul S. Heckbert. Surface simpli_cation using quadric error metrics. In SIGGRAPH '97 : Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pages 209_216, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[10] J.Y. Rau, L.C. Chen, F. Tsai, K.H. Hsiao, and W.C. Hsu. Lod generation for 3d polyhedral building model. pages 44_53, 2006.

[11] David Luebke. A survey of polygonal simpli_cation algorithms. Technical report, Chapel Hill, NC, USA, 1997.

[12] Taosong He, Lichan Hong, A. Kaufman, A. Varshney, and S. Wang. Voxel based object simpli_cation. In VIS '95 : roceedings of the 6th conference on Visualization '95, page 296, Washington, DC, USA, 1995. IEEE Computer Society.

[13] Jonathan Cohen, Amitabh Varshney, Dinesh Manocha, Greg Turk,

[14] Hans Weber, Pankaj Agarwal, Frederick Brooks, and William Wright. Simpli_cation envelopes. In SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 119_128, New York, NY, USA, 1996. ACM.

[15] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution

analysis of arbitrary meshes. In SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, pages 173_182, New York, NY, USA, 1995. ACM.

[16] Hugues Hoppe. Progressive meshes. In SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 99_108, New York, NY, USA, 1996. ACM.

[17] Tamal K. Dey and Jian Sun. De_ning and computing curveskeletons with medial geodesic function. In SGP '06 : Proceedings of the fourth Eurographics symposium on Geometry processing, pages 143_152, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[18] Oscar Kin-Chung Au, Chiew-Lan Tai, Hung-Kuo Chu, Daniel Cohen-Or, and Tong-Yee Lee. Skeleton extraction by mesh contraction. In SIGGRAPH '08 : ACM SIGGRAPH 2008 papers, pages 1_10, New York, NY, USA, 2008. ACM.

[19] Julien Tierny, Jean-Philippe Vandeborre, and Mohamed Daoudi. Analyse topologique et géométrique de maillages 3D pour l'extraction de squelette. In 19th French Chapter of Eurographics - 19èmes Journées de l'Association Française d'Informatique Graphique et de l'Association Chapitre Français d'Eurographics, pages 1_8, Bordeaux, France, 2006.

[20] Julien Tierny, Jean-Philippe Vandeborre, and Mohamed Daoudi. Graphes de Reeb de Haut Niveau de Maillages Polygonaux 3D. In COmpression et REprésentation des Signaux Audiovisuels (CORE-SA'06), pages 172_177, Caen, France, 2006.

[21] Grégoire Aujay, Franck Hétroy, Francis Lazarus, and Christine Depraz. Harmonic skeleton for realistic character animation. In SCA'07 : Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 151_160, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.