Triangle Mesh Compression and Homological Spanning Forests

Javier Carnero, Helena Molina-Abril, and Pedro Real

Computational Topology and Applied Mathematics Group,
Applied Mathematics I Department, University of Seville
javier@carnero.net, {habril,real}@us.es

Abstract. Triangle three-dimensional meshes have been widely used to represent 3D objects in several applications. These meshes are usually surfaces that require a huge amount of resources when they are stored, processed or transmitted. Therefore, many algorithms proposing an efficient compression of these meshes have been developed since the early 1990s. In this paper we propose a lossless method that compresses the connectivity of the mesh by using a valence-driven approach. Our algorithm introduces an improvement over the currently available valence-driven methods, being able to deal with triangular surfaces of arbitrary topology and encoding, at the same time, the topological information of the mesh by using Homological Spanning Forests. We plan to develop in the future (geo-topological) image analysis and processing algorithms, that directly work with the compressed data.

Keywords: Triangle Mesh Compression, Homological Spanning Forest, Computational algebraic topology.

1 Introduction

Polygon three-dimensional meshes have been widely used on many different applications to represent 3D objects. In fact, since triangles are the basic geometric primitives for standard graphics hardware and for many simulation algorithms, triangle meshes are the most commonly used. This is the reason why most of the effort in the field of static 3D model compression has been devoted to triangle meshes.

We focus here in triangle meshes that often require a huge amount of data for their storage, processing and transmission in the raw data format. Therefore, to find an efficient method for compressing these meshes is one of the aims of the work we present here.

Besides that, the ability of computing and storing topological information of these meshes is also an issue of interest for many problems dealing with them. This is the case of medical image processing, where topological information is crucial in order, for instance, to make a meaningful automatic classification of the images.

M. Ferri et al. (Eds.): CTIC 2012, LNCS 7309, pp. 108–116, 2012.

The method we propose here provides a lossless compression of the connectivity of the mesh, allowing the inclusion of its topological information in the coded file and/or its automatic computation in the decompression process.

In order to compress the connectivity of the mesh, a valence-driven approach started by Touma and Gotsman [2] is used. This valence-driven approach codifies the neighborhood of each vertex as a number following a certain order. Using this "valence"information the mesh can be reconstructed without loosing information. Our variation introduces an improvement over the currently available valence-driven algorithms, being able to deal with triangular orientable meshes of arbitrary topology (not necessary 2-manifold), keeping all the benefits of the valence-driven approach.

In the compression method we propose here, the data structures defined by Molina-Abril and Real [4] (called *Homological Spanning Forest*) can be easily computed without increasing the computational time of the algorithm. Let us notice that by using this structure, we encode not only basic topological information like Betti numbers, genus or Euler characteristic, but also advanced topological information (reconstruction of the boundary, homological classification of cycles, etc.). The inclusion of this structure in the compressed data provides a suitable framework for a geo-topological processing in the compressed domain (contractibility testing and transformability of cycles, topological analysis of ROIs, shortest path problems, etc.). The possibility of directly working with the compressed data, is an important advantage when dealing with large meshes and images.

The resulting algorithm uses less than 1.5 bits per vertex (bpv) on average to encode mesh connectivity. This compression ratio coincides with the state-of-the-art ratio that has not been seriously challenged till now. As other valence-driven algorithms, the proposed method can be used in progressive transmission, which means that the mesh can be decompressed, processed and rendered during the transmission process.

In particular the investigation about connectivity compression has been developed under the project *VirSSPA'10*, in the *Hospital Universitario Virgen del Rocío*, Seville (Spain), and financed by the *Consejería de Salud de la Junta de Andalucía* and *FEDER* founds.

In order to complete the goals of the project, an application developed in C++ implementing the introduced technique has been developed. A database of more than six hundred medical images of real patients from the *Hospital Universitario Virgen del Rocío* has been used to corroborate the theoretical results.

2 Connectivity Compression Method

For compressing the connectivity of an orientable triangle mesh, a valence-driven algorithm has been developed. This approach was presented in 1998 by Touma and Gotsman [2], compressing the connectivity in terms of the neighborhood of each vertex. In the algorithm we propose here, a random triangle of the mesh is selected to be the origin of three implosions (one for each of its three vertices).

Also the triangle face is defined as *the imploded region*, being its border defined as the three edges plus the three vertices.

The implosion of vertex v is defined as the process in which v and its neighborhood faces are destroyed by collapsing (or being squeezed in) on v. An implosion is always produced by a vertex in the border of the imploded region, growing it. The number of new vertices that have been added to the implosion region border due to the implosion is registered in order to be able to reconstruct the implosion in the decompression process.

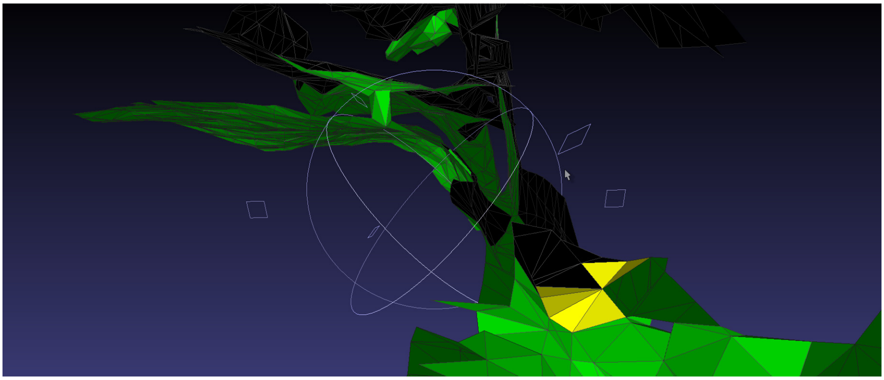

Fig. 1. An implosion (colored in yellow) from a vertex that belongs to a hole in the mesh (in green), and the growing of the imploded region (in black) surrounding the hole

This process also produces sorted list of vertices as they pass through the regular region of the mesh to the imploded region border. This distribution, based on the mesh topology, is very useful for the next step of a triangle mesh compression, that is the geometry compression (it will not be treated in this paper).

Two special cases of implosions can be distinguished when dealing with 3D triangle surfaces. Using the terminology coined by Touma and Gotsman, we call these two cases *split* and *merge*. Although these special cases are not exactly the same as those defined by Touma and Gotsman (as our method is developed from an implosion point of view), they both are produced by the same topological principles, so the notation has still sense in our case.

- SPLIT
  A *split* is produced when an implosion touches its own region border. This produces a split of the border into two, so if the imploded region has only one border, after the split it will have two.
- MERGE
  A *merge* occurs when an implosion touches another border that is different from the border where the vertex that produced the implosions belongs. In this case, a merge of the two borders needs to be done. If the imploded region has two borders, after the merge it will have only one.

3 Homological Spanning Forest Representation

Roughly speaking, topology in a discrete context helps to understand the degree of connectivity of subdivided geometric structures. For subdivided objects, homology is topology measured in terms of linear combinations (called chains) of unit elements or bricks (also called cells), and in terms of "boundary relations" describes the connectivity dependencies among these bricks. Homology depends on the ring of coefficient, and gives an algebraic answer in terms of formal sums of bricks that have no boundary (for example, closed subdivided curves or surfaces). These sums are called cycles and homology determines a representative cycle for each n-dimensional hole or homology generator the object has (connected components, tunnels, cavities, etc). In this way, homology can be considered as a specification of the contribution of each brick to the creation of the homology representative cycles.

In order to codify these connectivity information in an efficient way, we use here a graph representation called Homological Spanning Forest (or HSF for short). These hierarchical tree-like structure gives a positive and efficient answer to the problem of codifying and computing classical algebraic topological information (Euler characteristic, Betti numbers, classification and relations between cycles, etc.). A detailed explanation about the topological information that the HSF codifies, and its formal definition can be found in [5]. Relations between the HSF and Morse Theory, in [4].

We will not go here into details about the Homological Spanning Forest representation. An elementary example of a subdivided object is shown in Figure 2 for the understanding of this idea. Given a geometric graph G, its homology information can be directly captured by means of a spanning tree T of G. In fact, we transform T into a directed tree T^d by adding arrows to every edge in T, in such a way that at most one arrow comes out from each vertex. Therefore, there will be only one vertex s of G, called sink, from which no arrow comes out. In Figure 2 we interpret an arrow (e, f) in T^d from the vertex f to the vertex e as an elementary "deformation" operation, "contracting" in a continuous way the vertex f onto e through the edge (e, f) inside the object. The result of applying (no matter the order we choose) the set of homology-preserving operations represented by a red arrow in Figure 2, is a reduced structure consisting of only three bricks: the vertex e, and two loops or "edges" starting and ending at the same common vertex e (in fact, they represent the cycles $\{(c, e), (d, e), (c, d)\}$ and $\{(c, e), (c, f), (e, f)\}$ coming from (c, e) and (c, f), respectively).

The directed spanning tree T^d can be interpreted in dynamical terms, as the way in which the set of vertices of the graph is "collapsed" to a representative vertex of the connected component (in this case, the vertex e in black). These three representative cycles of homology generators (in this case, no matter of the ground ring we use but heavily dependent on the spanning tree T) are determined by the following bricks of G (called *critical*): the edges (c, f), (c, e) that belong to $G \setminus T$, and the sink vertex e that belongs T. Their integer homology groups are one copy of $\mathbb{Z}$ in dimension 0 and two copies in dimension 1. A Homological Spanning Forest representation $\mathcal{F}(G)$ for the subdivided geometric structure G is the set of trees $\mathcal{F}(G) = \{T^d, T_1, T_2\}$, where T_1 and T_2 are trees composed by only one "vertex": the original edges (c, e) and (c, f) respectively.

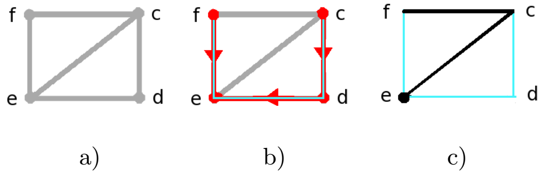

Fig. 2. a) A geometric graph G drawn on $\mathbb{R}^2$, b) a directed spanning tree (in red) showing a homological "deformation" process, and c) the minimal homological object (in black)

If we now add two triangles to the previous graph obtaining a new object O, we can now construct a tree T^{d_2} codifying the "collapsing" of (c, f) and (c, e) to the triangles (c, e, f) and (c, d, e) respectively (see Figure 3). Then, the HSF representation $\mathcal{F}(O)$ for the subdivided object O is the set of trees $\mathcal{F}(O) = \{T^d, T^{d_2}\}$.

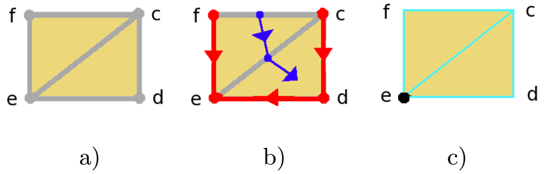

Fig. 3. a) A geometric object O, b) a directed spanning tree (in red) showing a homological "deformation" process, and c) the minimal homological object (in black)

This graph-based representation, suitably encodes advanced topological features of the object, due to the fact that the HSF forest can be automatically rewritten in algebraic terms (with coefficients in a field) as a chain homotopy operator, that determines a strong relationship at chain level (formal sums of bricks) between the geometric object and its minimal homological expression; that is a chain homotopy equivalence. The chain homotopy operator can be directly extracted from the forest as sums of cells following the paths of the trees (see [5] for more details). By advanced topological information we mean not only Euler characteristic and Betti numbers, but also classification of cycles, relations between cycles, etc.

4 Connectivity Compression Algorithm

The compression algorithm is presented in algorithm 1. The decompression method works similarly, but reading the implosions information from the compressed file and then creating the mesh.

Algorithm 1. Compression

```
Require: List of triangles and vertex information (coordinates, normals and color) of
    an orientable triangle mesh.
Ensure: Triangle mesh compressed.
Ensure: HSF of the mesh.
while there are no imploded vertices do
    Select a no imploded triangle.
    Define the triangle as imploded region.
    Build the HSF for that triangle.
    Define the triangle border as the imploded region border, and make it the active
    border.
    while there is an active border do
        if all vertices of the border have been imploded then
            Delete the border.
            if there is no inactive border then
                End loop.
            else
                Select an inactive border as active.
            end if
        else
            Select a not imploded vertex of the active border.
            Implode the vertex.
            Register valence, splits and merges.
            Build the HSF on the implosion, connecting them with the imploded region.
        end if
    end while
end while
return Homological Spanning Forest
return Compressed mesh
```

The Touma and Gotsman algorithm [2] has an average compression ratio of 1.5 bits per vertex. This ratio has not been seriously challenged till now. However, these results are purely empirical, and a theoretical study is not available.

Alliez and Desbrun [1] proposed a method to further improve the performance of Touma and Gotsman algorithm. They observed that the code produced by splits consumes a non-trivial portion of coding bits, and proposed some simple techniques to reduce it, especially for irregular meshes where this special code can be huge. Alliez and Desbrun proved that if the amount of the splits is negligible, the performance of their algorithm is upper-bounded by 3.24 bpv, which is exactly the same as the theoretical bits per vertex value computed by enumerating all possible planar graphs [6].

In the algorithm we present here, our results are close to the ones by Touma and Gostman. In the near future we plan to adapt the ideas presented by Alliez and Desbrun to our algorithm in order to reduce the compression ratio. This will give us an algorithm that is able to reach not only reach the theoretical minimal compression ratio, but also to include in the compressed file (by using the implosion technique) the topological information of the mesh (by computing the so called HSF data structures).

4.1 HSF Data Structures Generation

In [3], the HSF computation algorithm has quadratic time complexity. The advantage of including the HSF representation computation in the proposed compression method, is that the computational time is not increased. The forest can be directly computed at the same time the implosions are generated. On the same way, in case the compressed data do not contain the encoded HSF structure, they can also be computed during the decompression process or even directly from the compress domain.

The computation of the HSF is performed following a “star-shape” strategy (see Fig. 4). In this way when a vertex “v” implodes, the arrows going from its neighbors vertices to the vertex “v” are added to the HSF structure (colored in red in Figure 4). The first imploding vertex of the whole process is the sink vertex “s”. The arrows corresponding to the neighbor triangles are also added considering the star flow (colored in blue in Figure 4). Following this strategy, the trees.

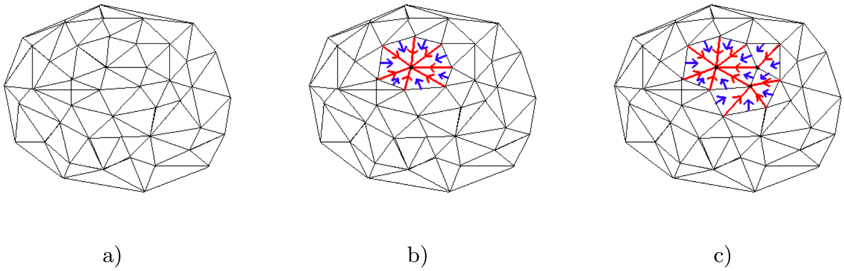

Fig. 4. a) The initial triangle mesh, b) The first implosion and HSF generation, and c) Next steps of the compression method and HSF generation

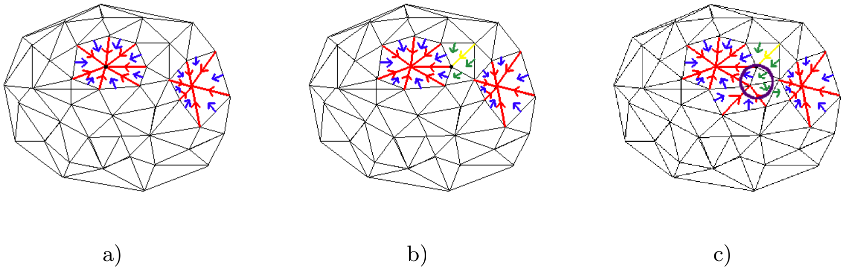

Fig. 5. a) Two imploded areas needing to be merged, b) The first step of the HSF representation computation in the merge operation, and c) The next steps of the HSF representation computation in the merge operation producing a bifurcation in the tree (marked within a circle in the picture)

Problems may occur when a merge or split takes place (see Figure 5). In this case, the HSF structure is constructed by making a bifurcation in the T^{d_2} tree.

5 Conclusions

In this paper we propose a lossless compression method based on a valence-driven approach. The main advantage of the method, is that it compresses triangle meshes of arbitrary topology and encodes, at the same time, the topological information of the mesh by using Homological Spanning Forests, without increasing the computational time.

The topological information encoded in the HSF structure can be later used for processing geometrical and topological information in the compressed domain (automatic cycle classification, how to transform -if it is possible- a cycle into another inside the object, recognition of 3D objects based on geometrical and topological features, etc). The algorithm allows progressive transmission, in the sense that the mesh can be decompressed, processed and rendered while its data is being received.

Acknowledgements. JC, HMA and PR acknowledge the support of the project VirSSPA'10 from the *Hospital Universitario Virgen Del Rocío* founded by *Consejería de Salud de la Junta de Andalucía* and FEDER Founds, the *Computational Topology and Applied Mathematics* CATAM research group FQM-296, and the Spanish MICINN Research Project MTM2009-12716.

References

1. Alliez, P., Desbrun, M.: Valence-driven connectivity encoding for 3d meshes (2001)
2. Chen, L., Georganas, N.D.: 3d mesh compression using an efficient neighborhood-based segmentation. In: Proceedings of the 9th IEEE International Symposium on Distributed Simulation and Real-Time Applications, DS-RT 2005, pp. 78–85. IEEE Computer Society, Washington, DC (2005)

3. Molina-Abril, H., Real, P.: A Homological–Based Description of Subdivided nD Objects. In: Real, P., Diaz-Pernil, D., Molina-Abril, H., Berciano, A., Kropatsch, W. (eds.) CAIP 2011, Part I. LNCS, vol. 6854, pp. 42–50. Springer, Heidelberg (2011)
4. Molina-Abril, H., Real, P.: Homological optimality in discrete morse theory through chain homotopies (2011); Elsevier Editorial System for Pattern Recognition Letters
5. Molina-Abril, H., Real, P.: Homological spanning forest framework for 2D image analysis (2011); Editorial System for Annals of Mathematics and Artificial Intelligence
6. Tutte, W.T.: A census of planar triangulations. Can. J. Math. 14, 21 (1962)