LOSSLESS AND SCALABLE 3D OBJECT CODING METHOD BASED ON MEDIAL AXIS TRANSFORMATION

Florent Dupont, Benjamin Gilles and Atilla Baskurt
LIRIS – Université Claude Bernard Lyon1
Bâtiment NAUTIBUS, 8 boulevard Niels Bohr
69622 Villeurbanne Cedex, France
fdupont@ligim.univ-lyon1.fr

ABSTRACT

3D discretized objects become more and more present in many applications for computer graphics. The need to store and transmit these data grows permanently. This paper introduces a new scheme of 3D object progressive transmission based on the coding of cone sections and balls from the medial axis and an optimization of the data transmission for a better scalability in terms of bit rates and quality. This representation of a voxelized object is a lossless approach and is very efficient for compression and progressive transmission, as shown in the results presented in this paper.

1. INTRODUCTION

Usually, polygonal meshes are used for visualizing 3D data. Several methods have been proposed on 3D surface mesh compression, mesh simplification and detail approximation in order to reduce the transmission time of 3D objects [6][7].

This paper deals with compression and transmission of 3D discrete data. For voxelized objects, skeletons and medial axis (MA) are often used in order to characterize shapes, and some works concern the generation of graphs to describe objects [8][10]. Borgefors and Nyström [4], Nilsson and Danielsson [9] present methods to have an efficient shape representation with MA. Giblin and Kimia [5] propose an hypergraph skeletal representation based on a local analysis of the MA. For the reconstruction part, Amenta [1] computes the interior of the union of balls with continuous methods.

In this study, we propose an efficient and scalable representation of a 3D object from the medial axis. Section 2 presents the MA computation based on the discrete distance map and the local maxima determination. We show in Section 3 how to optimize the number of points of the MA in order to reduce the transmission time on a network. We also present how to optimize the order of the transmitted data to have rapidly a good quality for the reconstructed object. In Section 4 we define cone sections and how to use them in order to reduce the number of transmitted elements. Section 5 presents the results for the evaluation of both the compression performance (the quantified results of the compression rates obtained compared to other methods) and the quality of the progressivity.

2. MEDIAL AXIS

The skeleton of a 3D object is defined as the set of voxels at equal distance from border. The MA has been introduced by Blum [2] and is very useful for object description and analysis. It is defined by the set of centers of maximal balls entirely included in the object. So, the MA is a skeleton containing for each point the distance from edge information.

Three main properties characterize MA: homotopy preservation, good localization (centered on the object) and reversibility. The last one guarantees an exact representation of the object after transmission and reconstruction and is essential for lossless compression.

2.1 Discrete Distance Map

To compute the MA, one way is to obtain first a discrete distance map where each voxel of the 3D object has the value of the minimal distance from the edge [3]. This method does not guarantee the connectivity preservation but this point is not important for a compression purpose. Discrete distances are considered [11] and they are based on 3D voxels neighborhoods. We have implemented four different configurations that lead to four different distances: D6, D18, D26 and chamfer distance with 3-4-5 parameters.

2.2 Local Maxima

The MA is the set of centers of the maximal balls included in the object and is obtained from the distance map by computing the local maxima in the neighborhood considered with the chosen distance.

3. MEDIAL AXIS OPTIMIZATION

3.1 Medial Axis Reduction

The MA is not minimal and contains redundant points. Some balls can be eliminated without changing the reconstructed object after transmission. The aim of our optimization method is to suppress from MA, points that are not indispensable while preserving the exact reconstructed object. Our method consists in computing for each MA point the number of voxels that belong to the associated ball and not to others. We call this characteristic the "intrinsic volume". The list of MA points is sorted by increasing radius. Then the intrinsic volume is computed for the first point of the list. If this value is null the point called « Zero volume » is suppressed from the list. This procedure is repeated while there are still unprocessed points. Characteristics of all points have to be updated at the end of the algorithm because of their mutual influence. The sorting stage allows a huge reduction of the number of points from the MA (small balls are preferentially suppressed from the MA).

3.2 Progressive Transmission Optimization

The optimized MA is a list of points (X,Y,Z coordinates) with associated radius. The list can be sorted in order to achieve rapidly a good quality of the reconstructed object. The first transmitted point is associated to the highest radius. Then, two strategies can be employed by sorting the list of points either by decreasing radius, or by decreasing intrinsic volume. These two methods are compared in Section 5.

The sorted list of points is prepared for transmission by coding the coordinates and radii in a binary file with the minimum number of bits dedicated to each coordinate and radius. A more efficient entropic coder could be implemented to reduce the size of the transmitted file.

3.3 Progressive Transmission Scheme

The two optimization phases described in the previous section are very important for the progressive transmission aim. They are successively applied on data. The MA computation and the optimization stages can be computed for all the norms (Section 2) and the better one can be retained. This leads to the smallest binary file even if computation time is higher (compression can be processed offline).

4. CONE SECTIONS

During the reconstruction, other primitives than balls can be used. Here, we propose a reversible method that uses particular primitives to decrease the size of data whilst ensuring exact reconstruction. To improve the former method, i.e. to delete more MA points, the idea consists to link the balls by cone sections. We define cone sections as the set of the balls between two balls where radius are linearly interpoled (Figure 1).

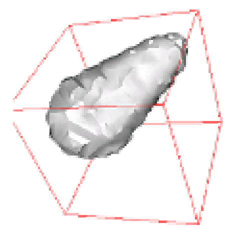

Figure 1. An example of cone section with chamfer metric

For tubular objects, the number of MA points can be very large. However, the number of cone sections to decompose them exactly is generally small, i.e. for a cylinder, we only need one cone which links two extremal balls. This point has been well depicted by Reinders, and al [10]. Our algorithm, by taking as the input the MA balls, intents to select the best links between these balls in order to describe the object in the most compact way. The computation of the optimal solution would be too heavy as the number of combinations to look at is almost infinite. Hence, we assumed that decreasing the most the distortion/time curve of the transmission for each transmitted level of detail (LOD) would give a good solution.

The list of cone sections which are totally included in the object (to avoid errors during the reconstruction) is exhaustive: every pair of MA balls is tested to see if the cone section linking them belongs to the object. Then, volumes of primitives (balls and cones sections) are computed, in number of voxels, and stocked into a list. It is an initialization of the next step. The loop aims to select, for each iteration, the primitive that brings the biggest volume to the object, until the object is totally reconstructed. The selection is made by taking the biggest volume from the volumes list. As soon as a primitive is selected, its volume has to be subtracted from the others. So the list is updated and then corresponds to the list of added volume. Furthermore, when it contains only null volumes, it means that any voxel cannot be added to the object; the union of selected primitives is at that time exactly the object.

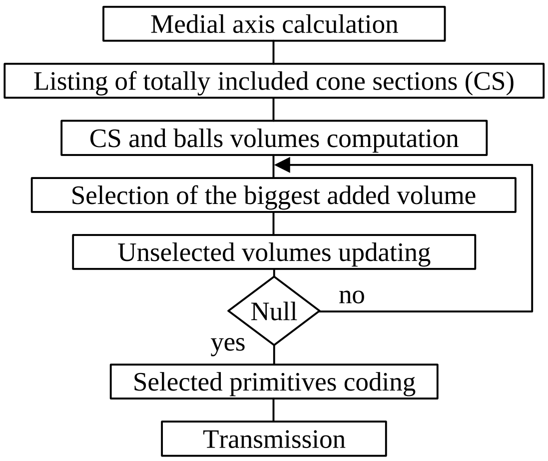

Figure 2. Coding scheme of the "cone sections" method

The coding of the list of primitives is quite straightforward: besides balls coordinates and radius coding, links between balls are represented by relative indexes. Actually, the use of relative indexes is far more efficient (in terms of size) than the use of absolute ones because relative indexes distribution is not uniform at all. We have chosen a particular binary coding for indexes in order to reduce the size of needed data. In addition, we used 1-bit separators between indexes to handle the fact that a ball can have any number of links.

5. RESULTS

Results given in this section were obtained with different volumes. We expect two kinds of results from this study: the compression efficiency with the comparison to standard encoders and the quality of the progressive method.

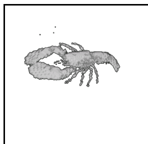		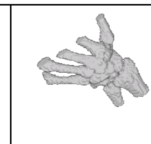	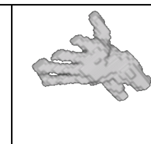
Lobster	Vertebra	Hand_1	Hand_2
77 x 83 x 16	66 x 66 x 66	37 x 63 x 66	19 x 32 x 34

Table 1. Test images

5.1 Data compression

After coding balls and/or cone sections, the size of the binary file obtained can be compared to the one given by classical encoders like GZIP and JBIG. We have implemented the progressive 2D compression scheme JBIG in 3D by considering a volume as a big 2D image with a juxtaposition of 2D slices. A drawback is the necessity to have at least one dimension as a power of 2.

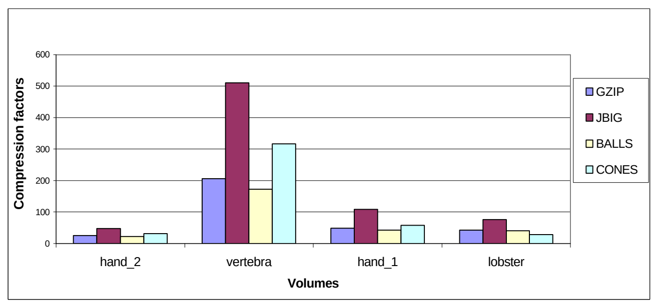

Figure 3. Compression factors

Figure 3 shows the compression factors (the size of raw file over the size of compressed one) for our methods (the best norm is used for each object), also for GZIP and JBIG. For the compression of natural objects, our methods are a bit weaker than the two others, especially for thin objects. However, they are most interesting because they present progressive transmission functionality of 3D objects.

To compare with a mesh approach, we have meshed the volumes and compressed them with GZIP (without real mesh compression at this time). Size of files are given in table 2. Our method provides smaller files (ratio equal to about 70) since it is well adapted to voxelized data. The counterpart is that it is necessary to compute meshes after transmission in order to visualize 3D objects.

	Lobster	Vertebra	Hand_1	Hand_2
Mesh (bytes)	321294	43479	53127	190937
Cones (bytes)	3641	909	658	2675
Cones (bit/voxel)	0.28	0.025	0.25	0.139

Table 2. Size of compressed files

5.2 Progressive transmission

Figure 4 shows the distortion's evolution in function of the percentage of transmitted data, which is proportional to the time at a fixed network's rate. The distortion is the error on the reconstructed volume from received data in comparison to the original one. Distortion is calculated using the metric adopted by MPEG–4:

D = (nb of pixels in error) / (nb of pixels into object)

In Figure 4, the corresponding curves for the JBIG method, for our "balls" and for our "cone sections" methods are shown. The point called "100%" corresponds to the total transmission of the file obtained with the "cones" method. With our "cones" method, even if the transmission is a bit longer than JBIG in some cases, we have a good approximation of the volume earlier (at about 20% of transmitted data).

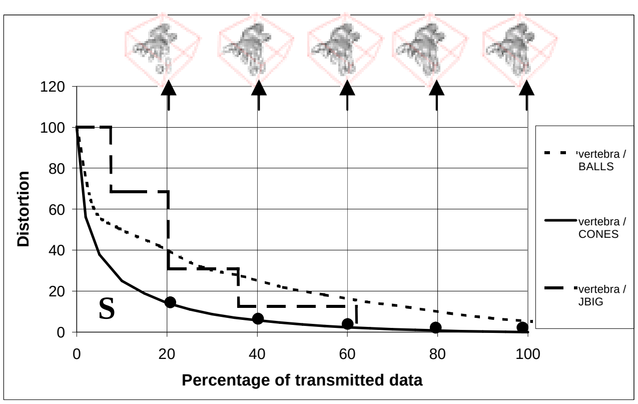

Figure 4. Distortion/time curves

To compare these two methods (and the JBIG one as well), a good criterion is to consider the area under the distortion/time curve (S in Figure 4). As we want to reduce the transmission time and the distortion, we want to minimize this area. The quality of the method is calculated with this criterion and results are presented in Figure 5.

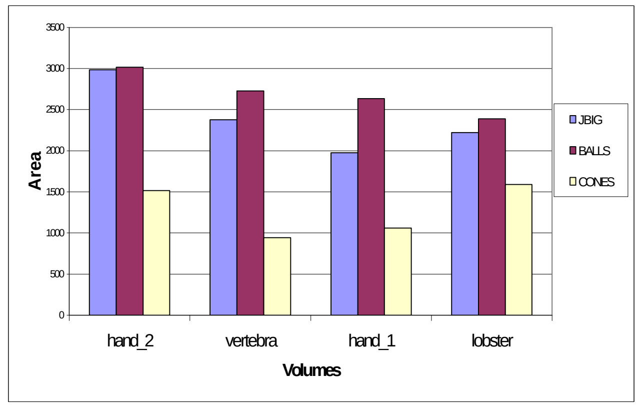

Figure 5. Area comparison

In all cases, our "cones" method gives better results than JBIG and our "balls" method. A reconstruction sequence with the "cones" method is given in figure 6.

6. CONCLUSION AND FUTURE WORK

In this paper, we have introduced a new method to transmit progressively 3D binary volumes. We have demonstrated that our method is able to give quickly a good approximation of a 3D object to the end-user. For complex volumes, the compression rates are better than the ones obtained with classic encoders and mesh compression. The use of data which contains 3D topological information will permit to compute 3D transformations and geometric properties easily. It opens perspectives for volume animation using transformations applied on MA and for indexing and retrieval volumes in large databases.

7. ACKNOWLEDGMENTS

Thanks to Florence Denis for the 3D hand image.

8. REFERENCES

[1] N. Amenta, R.K. Kolluri, "The medial axis of a union of balls", Computational Geometry, Vol. 20 No. 1-2, 2001, 25-37.

[2] H. Blum, "A transformation for extracting new descriptors of shape", In Symposium on Models for the Perception of Speech and Visual Form, M.I.T. Press, 1964, 139-146.

[3] G.Borgefors, "Distance Transformation in Arbitrary Dimension"s, Computer Vision, Graphics and Image Processing, Vol. 27, 1984, 321-345.

[4] G. Borgefors, I. Nyström, "Efficient shape representation by minimizing the set of centres of maximal discs / spheres", Pattern Recognition Letters, Vol. 18, 1997, 465-472.

[5] P. Giblin, B.B. Kimia, "A Formal Classification of 3D Medial Axis Points and their Local Geometry", CVPR00, 566-573.

[6] A. Guéziec, G. Taubin, F. Lazarus and W. Horn, "Simplicial Maps for Progressive Transmission of Polygonal Surfaces", VRML'98, ACM, February 98, 25-31.

[7] H. Hoppe, "Progressive Meshes", SIGGRAPH 1996, 99-108.

[8] R. Kresch and D. Malah, "Skeleton-Based Morphological Coding of Binary Images", IEEE Trans. Image Processing, Vol. 7, No.10, Oct. 1998, 1387-1399.

[9] F. Nilsson and P-E. Danielsson, "Finding the Minimal Set of Maximum Disks for Binary Objects", Graphical Models and Image Processing, 59(1), Jan. 1997, 55-60.

[10] F. Reinders, M.E.D. Jacobson, F.H. Post, "Skeleton Graph Generation for Feature Shape Description", Proc. Data Visualization 2000, 73-82.

[11] Remy, E., and Thiel E., "Medial Axis for chamfer distances: computing look-up tables and neighborhoods in 2D or 3D", Pattern Recognition Letters, Volume 23, Issue 6, April 2002, 649-661.

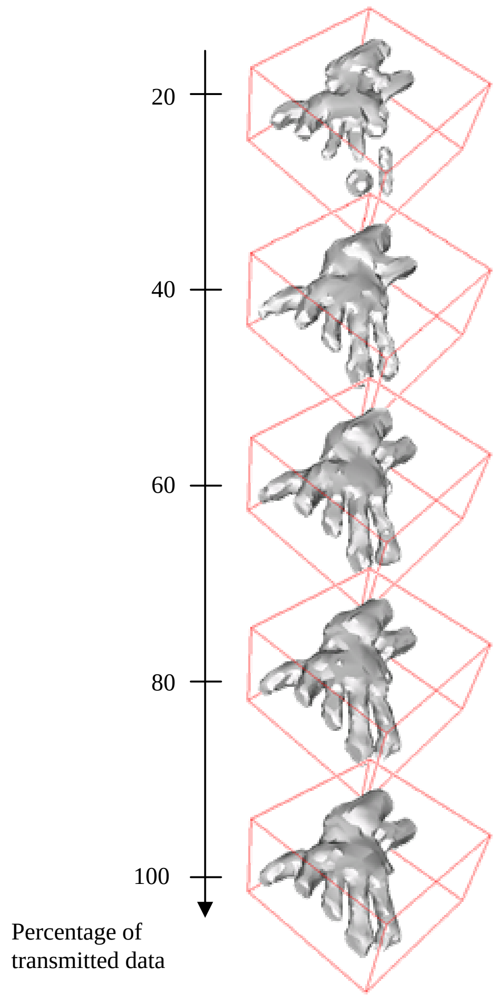

Figure 6. Reconstruction sequence with "cones"