

Octree Based Compression of Volumetric and Surface 3D Point Cloud Data

Rizwan A Siddiqui¹, Işıl Celasun.¹, UluğBayazit.²,

¹ Istanbul Technical University Istanbul

² Isık University Istanbul

Abstract. In this paper we have introduced a novel technique for the geometry compression of the point cloud data. The compression can be deemed lossless and it works on volumetric as well as surface 3D data. Our method is based on an Octree decomposition of space. The point cloud is encoded in terms of occupied octree cells. The octree is generated with rate distortion constraint. To compress the octree we have first employed the run length coding which is followed by Huffman coding technique.

Keywords: Octree, 3D Delaunay triangulation, Rate Distortion techniques.

1 Introduction

3D graphics data are widely used in multimedia applications such as video gaming, engineering design, virtual reality, e-commerce, medical and scientific visualizations. With the increasing capability of 3D data acquisition devices and computing machines, it is relatively easy to produce digitized models with millions of points. The increase in both availability and complexity of 3D digital models makes it critical to efficiently compress the data so that they can be stored, transmitted, processed and rendered efficiently.

Conventionally, polygonal meshes have been widely used to represent 3D objects. 3D mesh is composed of three parts: geometry, topology and its attributes. Therefore, while compressing the mesh data all the three have to be coded efficiently. In recent years, point based 3D model representation gains more popularity with several advantages. For example, the triangulation overhead is saved, processing and rendering are facilitated without the connectivity constraint and objects of complex topology can be more easily represented. These factual arguments make the point based representation an ideal choice in many applications that use high quality 3D models.

3D data, in fact, is divided into two main classes: the surface and the volumetric data. Most of the efforts are made in the past to compress the 3D surface data and very little attention is given the volumetric form of the 3D data.

In this work, we propose a novel encoder and decoder which performs equally well on surface as well as volumetric 3D data. The encoder is based upon rate-distortion controlled octree encoding which process point samples of 3D objects with arbitrary

topology and codes them. The output of octree encoder is then first run-length coded, followed by Huffman coding which lead to greatly reduced entropy values and as a result, significant improvement over the prior art in coding efficiency. The Delaunay triangulation based method is finally used for rendering the objects. The reason being they have a robust structure and their counter part Voronoi diagram spaces may be used for watermarking as future work.

1.1 Related Work

Several schemes for point sampled geometry have been proposed. Fleishman et al [1] proposed the progressive point set surfaces, which were based on the MLS surface definition put forwarded by Alexa et al [2]. Unfortunately the method has a tendency to smooth out the sharp features as a consequence of the MLS surface approximation. The algorithm of Ochotta and Saupe [3] makes use of the MLS surface as well. They use a set of planar height fields to resample the surface and encode the data using image based technique. Huang et al [4] proposed the progressive coder which encodes the geometry and attributes in the unified framework. Gumhold et al [5] proposed a prediction tree for each input model to facilitate prediction and entropy coding. To the best of our knowledge none of the proposed coding system have taken into account both surface and volumetric compression simultaneously. Isenberg et al [6] have proposed a method of streaming compression of tetrahedral meshes which takes into account the compression of volumetric data in mesh form.

2 Octree Compression

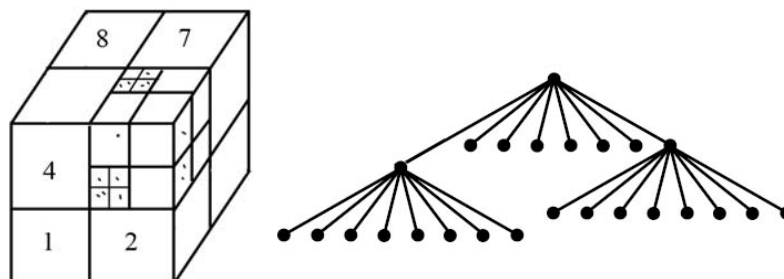


Fig 1. Structure of octree.and the division of the octant

The octree-based geometry compression as introduced by Botsch [7] and Peng [8] in their work. An octree structure can be defined by enclosing the domain of interest in a cube which represents the root of the octree, and then subdividing the cube into the eight octants of the root by bisection in all three directions. Those octants are then recursively subdivided to whatever levels are desired. The terminal octants of that subdivision process represent the basic units referred to by the application using the octree. See Fig. 1 for a picture of an octree and its corresponding data representation as a tree. The octree we have implemented had two constraints imposed over it.

1. One Point Occupancy
2. Rate Distortion Constraint.

2.1 One Point Occupancy

The octree encoder keeps dividing itself till there is only one point is present in each octant. This method ensures that there is no loss of data. Although the reduction of data facilitates the progressive reduction of data which may be utilized in the progressive transmission but in this work the data is controlled to remain same. This results in the formation of lot of octants which do not possess any point inside them. For efficient coding, a 1-bit flag is used to signify whether a child cell is empty or not, with '1' indicating a non-empty child cell and '0' an empty child cell. If we traverse all child cells according to a fixed order, and collect the flag bits of all child cells, we will obtain an 8-bit code, which we call the octree output code that has to be compressed.

2.2 Rate Distortion Constraint

When all the child cells have just one point in its octant then this method is employed to decrease the distortion which appears in the case of Octree implementation. Obviously this further division will give rise to seven non empty and one filled cell. So avoid the greedy division a constraint of rate has been employed over the division. Given a octree T , let \tilde{T} stands for its leaves (or terminal nodes). Assume that we split $t \in \tilde{T}$ into eight new child cells. Let D and R stands for the distortion and rate, respectively, measured by T , and D' and R' stand for the distortion and rate after t is split. Let

$$\Delta D = D' - D \quad (1)$$

and

$$\Delta R = R' - R \quad (2)$$

be the change in the distortion and rate respectively, due splitting t . The magnitude of the ratio of the change in distortion to change in rate (and hence the goodness of split for the cell t) is just

$$\lambda = -\frac{\Delta D}{\Delta R} \quad (3)$$

3 Index Compression

The index generated by the octree encoder now can be coded into many different ways. Schnabel and Klein [9] have employed arithmetic coding scheme. But in this work we have first employed the run length coder over the octree output array. The reason being, since at the end of our output there was a big sequence of string with all zeros in it. Since 0 in our coding scheme also indicates that the octant has not to be

divided any further. After the application of the run length coder, the resultant array was Huffman coded to exploit the redundancy within the output. The resultant was at a very low level of distortion we were able to code our data under 3 bit per cell.

4 Experimental Results.

Three point based models are tested in our experiment. They are skull, Torso and Stanford Bunny. The Torso data was obtained from Utah University repository (http://software.sci.utah.edu/cgi-bin/archive_entry.pl?arch=ncrrdata&soft=scirun), the skull data was obtained from aim shapes (<http://shapes.aim-at-shape.net>) and the Stanford Bunny was obtained from Stanford repository. The Torso and the skull were the volumetric data where as the Bunny is as well known a surface data. To compare the performance of our algorithm we have also implemented the work of Isenberg et al [6] which is utilized for the volumetric tetrahedral mesh compression. Since our aim was to develop an algorithm which performs equally well on Volumetric as well as surface 3D data.

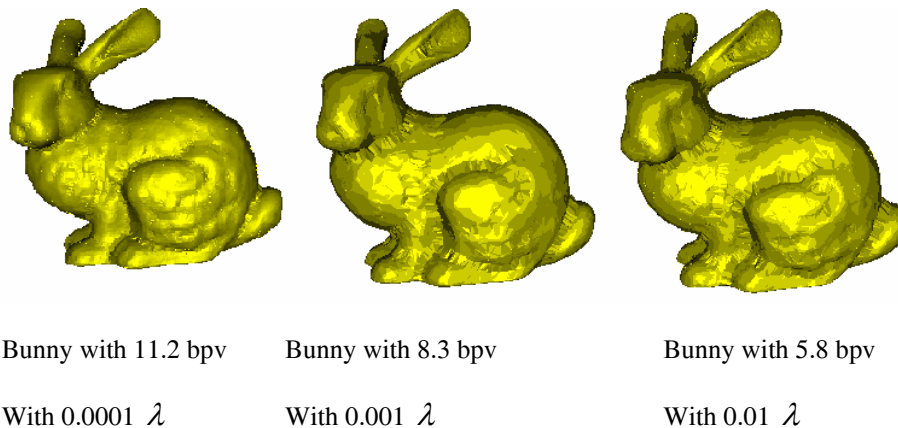
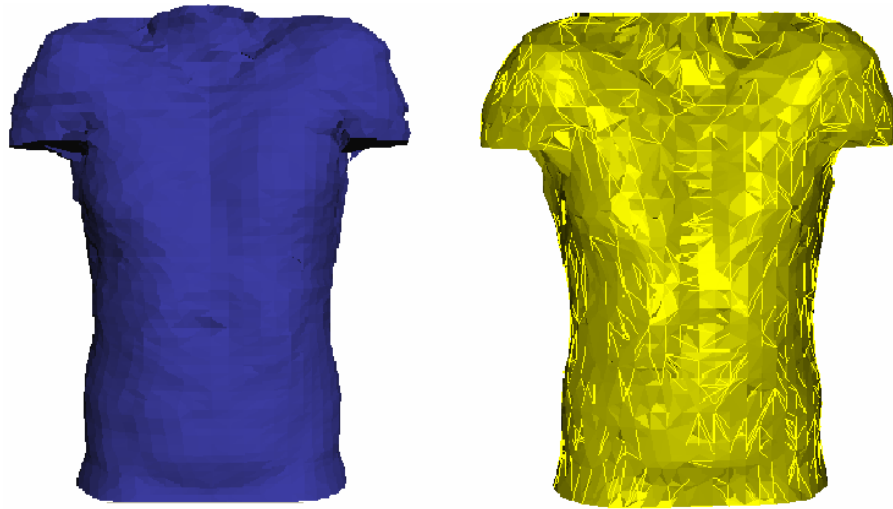


Fig 2 Different states of Bunny data being shown.

The Figure 2 clearly manifests the result of increasing lambda which was derived in equation (3) of section 2. The decrease in the lambda causes the introduction of more details into the data. As we can observe that with the lowest amount of lambda at the extreme left of the figure the detailed surfaces of ear are very visible but with the degradation in the lambda values although there is a bit saving observed but the details are lost. For comparison the proposed algorithm has been employed over the Torso volumetric data with 168,930 points. The coding by Isenberg et al resulted in around 25bpv where as according to the proposed algorithm as shown in the right hand side of the figure is 16bpv.



The [6] Isenberg's algorithm for Torso

Proposed algorithm applied over Torso

Fig 3 The Visual results for torso data for comparison.

Although the algorithm proposed by Isenberg et al [6] is superior from the proposed algorithm in terms of time and streaming compression but if geometry compression of meshes is considered than our algorithm performs better.

References

1. Fleishmann, S., Cohen-or D., Alexa, M., Silva, C.T.: Progressive Point Set Surfaces. *ACM Trans. Graph.* 22, 4(2003), 997-1011.
2. Alexa, M., Behr, J., Cohen-or D., Fleishmann, S., Levin, D., Silva, C.T.: Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*9, 1(2003), 3-15.
3. Ochotta, T., Saupe, D.: Compression of Point Based 3D Models by Shape-adaptive Wavelet Coding of Multi-height Fields. In *Proceedings of the Eurographics Symposium on Point Based Graphics (June 2004)* pp. 103-112.
4. Huang, Y., Peng, J., Kuo, C-C.J., Gopi, M., Octree-Based Progressive Geometry Coding of Point Clouds. In *Proceedings of the Eurographics Symposium on Point Based Graphics (2006)*
5. Gumhold, S., Karni, Z., Isenberg, M., Seidel H.-P.: Predictive Point-Cloud Compression. In *Siggraph Sketches (2005)*.
6. Isenberg, M., Lindstrom, P., Gumhold, S., Shewchuk, J., Streaming Compression of Tetrahedral Volume Meshes appeared in *Graphics Interface (2006)* 1-7.
7. Botsch M., Wiratanaya A., Kobbelt L.: Efficient high quality rendering of point - sampled geometry. In *EGRW '02: Proceedings of the 13th Eurographics workshop on Rendering (Aire-la-Ville, Switzerland, Switzerland, 2002)*, Eurographics Association, pp. 53-64.
8. Peng J., Kuo C.-C. J.: Geometry-guided progressive lossless 3D mesh coding with octree (OT) decomposition. In *ACM SIGGRAPH (2005)*, pp. 609-616.
9. Schnabel, R., Klein, R.: Octree Based Point Cloud Data Compression In *Proceedings of the Eurographics Symposium on Point Based Graphics (2006)*