

# CONNECTIVITY-GUIDED ADAPTIVE LIFTING TRANSFORM FOR IMAGE LIKE COMPRESSION OF MESHES

Kıvanç Köse<sup>1</sup>, A. Enis Çetin<sup>1</sup>, Uğur Güdükbay<sup>2</sup>, Levent Onural<sup>1</sup>

<sup>1</sup>Department of Electrical and Electronics Engineering, <sup>2</sup>Department of Computer Engineering  
Bilkent University, 06800 Bilkent, Ankara, Turkey

## ABSTRACT

We propose a new connectivity-guided adaptive wavelet transform based mesh compression framework. The 3D mesh is first transformed to 2D images on a regular grid structure by performing orthogonal projections onto the image plane. Then, this image-like representation is wavelet transformed using a lifting structure employing an adaptive predictor that takes advantage of the connectivity information of mesh vertices. Then the wavelet domain data is encoded using “Set Partitioning In Hierarchical Trees” (SPIHT) method or JPEG2000. The SPIHT approach is progressive because the resolution of the reconstructed mesh can be changed by varying the length of the 1D data stream created by the algorithm. In JPEG2000 based approach, quantization of the coefficients determines the quality of the reconstruction. The results of the SPIHT based algorithm is observed to be superior to JPEG200 based mesh coder and MPEG-3DGC in rate-distortion.

**Index Terms**— 3D model compression, image-like mesh representation, connectivity-guided adaptive wavelet transform.

## 1. INTRODUCTION

Multiresolution representations can be defined for 3D meshes. It would be desirable to obtain the coarse representation from the fine representation using computationally efficient algorithms. Wavelet-based approaches are applied to meshes to realize a multiresolution representation of a given 3D object.

There exist several mesh compression algorithms in the literature [1, 2, 3]. They can be classified as *progressive mesh compression* and *single-rate compression*. In single rate compression schemes, the mesh data is compressed before the transmission and then all the data is sent. The data is decodable if all of it is received. Progressive mesh representations enables the user to obtain different resolutions of the model using different sizes of code stream. Therefore, first a low-resolution data is decoded and then the decoded model is updated to a higher resolution using new-coming data.

In this paper, an adaptive wavelet-based mesh compression framework is proposed. The proposed compression method uses Set Partitioning In Hierarchical Trees (SPIHT) [4, 5] or JPEG2000 [6] to encode the wavelet domain mesh data. First 2D images are obtained from the mesh. 3D vertex data are projected on regularly sampled 2D grids in such a way that some nodes of the grid have a specific value related to the actual positions of the vertices in the 3D space. The proposed framework for mesh compression is summarized in Figure 1. During *mesh-to-image transform* stage of the proposed

method the positions of the vertices are quantized so that the data becomes more regular.

The transformation steps are used for converting the 3D mesh to a representation on which image-signal processing methods can be applied. Two popular approaches are (i) parameterizations of meshes [7] and (ii) using modified versions of signal processing algorithms, like relaxation [8]. The first approach is not practical for complex models since it requires the solution of several linear equations. The second approach is relatively easy to apply since it only needs the signal processing tools to be adapted to the mesh data. Our algorithm first converts the data to a 2D image by a transformation whose complexity does not depend on the model complexity and then uses image processing algorithms.

This paper is organized as follows. The techniques used in the implementation of the mesh compression framework are explained in Section 2. The simulation and compression results are presented in Section 3. Conclusions are given in Section 4.

## 2. COMPRESSION FRAMEWORK

The stages of our mesh compression algorithm and adaptivity that we embed in the wavelet transform, are explained in this section.

### 2.1. Initialization and Projection

3D mesh data is formed by geometry and connectivity information. Geometry information of the 3D mesh contains the coordinates of the mesh vertices  $S = \mathbf{s}_i, i = 1, \dots, M$ , where  $M$  is the number of vertices in the mesh. Let us define our 3D space as  $\mathbf{X}' = (x', y', z')^T$ . Thus, the coordinates of the mesh vertices are given as,

$$\mathbf{s}'_i = (x'_i, y'_i, z'_i)^T, i = 1, \dots, M. \quad (1)$$

In our approach we first normalize the space such that all vertices are in  $R^3[-0.5, 0.5]$  as,

$$\mathbf{X} = (x, y, z)^T = \alpha \mathbf{X}', \alpha \text{ is a constant.} \quad (2)$$

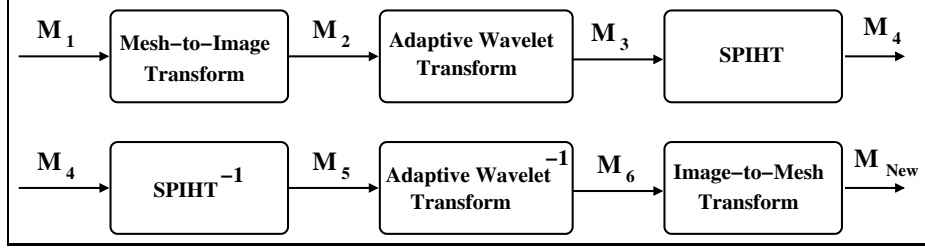
Let the vertices of the normalized mesh be

$$\mathbf{s}_i = (x_i, y_i, z_i)^T, i = 1, \dots, M. \quad (3)$$

Then the selected projection plane  $\mathbf{P}(u, v)$  is discretized using the sampling matrix  $\mathbf{V}$ . Sampling matrix  $\mathbf{V}$  is defined as,

$$\mathbf{V}_{rect} = \begin{pmatrix} T & 0 \\ 0 & T \end{pmatrix} \text{ for rectangular sampling,} \quad (4)$$

$$\mathbf{V}_{quinc} = \begin{pmatrix} T & T/2 \\ 0 & T/2 \end{pmatrix} \text{ for quincunx sampling.} \quad (5)$$



**Fig. 1.** The proposed framework for mesh compression.  $M_i$  represents the mesh data at stage  $i$ .

For triangular meshes, the settlement of mesh vertices in 3D space fits better to quincunx sampling than rectangular sampling [9]. Thus, using quincunx sampling gives good approximations for mesh vertices.

Let  $\tilde{s}_i$ , which is a 2D vector, be the projection of  $s_i$  onto the plane  $P$ . Furthermore, let  $\tilde{d}_i$  be the perpendicular distance of  $s_i$  to the plane  $P$ . The projection plane  $P$  is selected as one of the  $xy, xz, yz$  planes. The mesh is first projected on all of these planes and the one which has the maximum number of projected vertices is selected as the projection plane.

The vertices that can be assigned to a grid point  $\mathbf{n} = [n_1, n_2]$  forms a set of indices  $J_{n_1, n_2}$  defined by,

$$J_{n_1, n_2} = \left\{ j \mid |\tilde{s}_j - \mathbf{V}\mathbf{n}| < T/2 \forall n_1, n_2 \right\}, \quad (6)$$

where  $\mathbf{V}$  is a sampling matrix and  $n = [n_1, n_2]$  represent the indices of the discrete image as shown in Figure 2. Sampling matrix  $\mathbf{V}$  determines the distance between neighbouring grid points.  $\mathbf{V}$  can be defined as a quincunx or a rectangular sampling matrix [9]

Then the 3D mesh is transformed to two 2D image-like representations by:

$$\mathbf{I}_1[n_1, n_2] = \begin{cases} \tilde{d}_i, & \text{any } i \in J \\ 0 & \text{otherwise.} \end{cases}, \quad (7)$$

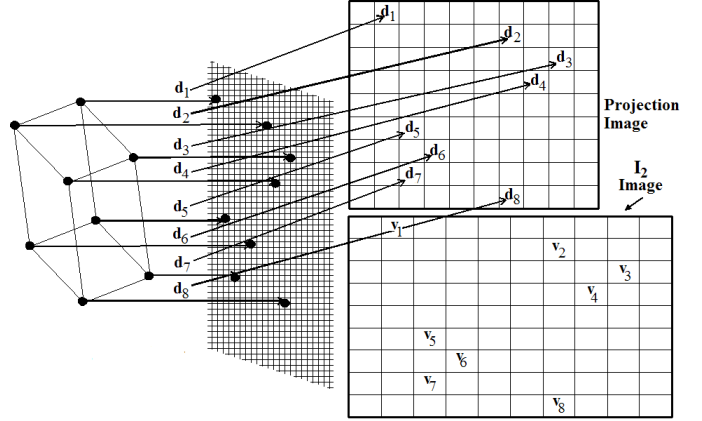
$$\mathbf{I}_2[n_1, n_2] = \begin{cases} i & \tilde{d}_i = \mathbf{I}_1[n_1, n_2] \\ 0 & \text{otherwise} \end{cases}. \quad (8)$$

The first image stores the perpendicular distances of the vertices to the selected planes and the second image holds the indices of the vertices. The first channel images are then wavelet transformed and SPIHT encoded. The second channel image takes its pixel values from vertex indices. Then it is converted to a list of indices. This list is differentially coded and sent to the other side.

Using Eqns. 7 and 8 a pixel-vertex correspondence for each vertex is tried to be found. However sometimes more than one vertex have the same projection in the image-like representation. Therefore, one of the vertices is chosen for the calculation of the pixel value and the others are discarded. By increasing the number of projection planes or decreasing the sampling frequency by changing the sampling matrix  $\mathbf{V}$  we can handle more vertices.

Both methods increase the reconstruction quality since they decrease the number of lost vertices. However, they will lead to a decrease in the compression ratio. In our approach we used one densely sampled plane. The recovery of lost vertices is handled by connectivity - based interpolation, which is explained in Section 2.4.

The projection operation and the image-like representation of a simple 3D object is shown in Figure 2. Pixel values are the distances of the object vertices from the projection plane. Most of the image contains empty grid points.



**Fig. 2.** Illustration of the projection operation and the resulting image-like representation.

## 2.2. Connectivity-Guided Adaptive Wavelet Transform

Unlike natural images there may be no correlation between neighboring pixels in our image-like mesh representation since neighboring pixels may not be coming from neighbouring vertices of the mesh. Thus, instead of predicting non-zero pixels in our representation from their neighboring non-zero pixels, an adaptive wavelet transform scheme which makes predictions using connectivity information is introduced here.

$\mathbf{I}_1[n_1, n_2]$  is the first channel image that stores the perpendicular distance between the corresponded vertex - pixel pair and  $\mathbf{I}_2[n_1, n_2]$  stores the vertex indices. The wavelet lifting transform is implemented in a separable manner. Thus, both  $\mathbf{I}_1$  and  $\mathbf{I}_2$  are polyphased in the horizontal direction as,

$$\begin{aligned} \mathbf{I}_a[n_1, n_2] &= [\mathbf{I}_{a1} | \mathbf{I}_{a2}] \\ \mathbf{I}_{a1}[n_1, n_2] &= \mathbf{I}[n_1, 2n_2] \\ \mathbf{I}_{a2}[n_1, n_2] &= \mathbf{I}[n_1, 2n_2 + 1], \quad a = 1, 2. \end{aligned} \quad (9)$$

Thus,  $\mathbf{I}_{22}[n_1, n_2] = i, i \in \{1, \dots, M\}$ . Using connectivity information we find a list of neighbors  $nlist(j), j = 1, \dots, M$  that holds the indices of the vertices connected to the vertex with index  $j$ . The predictions for  $\mathbf{I}_{12}[n_1, n_2]$  values are done using this  $nlist_{valid}$  list, which is defined as,

$$nlist_{valid}(j) = nlist(j) \cap \mathbf{I}_{21}[n_1, n_2]. \quad (10)$$

List of vertex indices that are on image  $\mathbf{I}_{22}$  is  $list_{22}$ . For each element  $k$  of  $list_{22}$ , a prediction should be found from  $\mathbf{I}_{11}$  image. Valid neighbors of  $k$  can be found using Equation 10. So the prediction of vertex  $k$  is defined as

$$\mathbf{I}_{k \text{ pred}} = \frac{\sum_m (\mathbf{I}_{11}[n_1, n_2])}{m}, \quad (11)$$

where  $\mathbf{I}_{21}[n_1, n_2] \in nlist_{valid}(k)$  and  $m$  is the number of the elements of  $nlist_{valid}(k)$ . Then  $\mathbf{I}_{12}[n_1, n_2]$  is updated as,

$$\mathbf{I}_{new12}[n_1, n_2] = \mathbf{I}_{12}[n_1, n_2] - \mathbf{I}_{k \text{ pred}}, \text{ where } \mathbf{I}_{22}[n_1, n_2] = k. \quad (12)$$

If no valid neighbors exist for a vertex, no estimation is carried out. Otherwise, a prediction is made for the value of the pixel. The same procedure is applied in the vertical direction of the low pass part of the image. The high pass part is only polyphased using lazy filters. Four small images are obtained at the end of wavelet transform. The inverse of the connectivity-guided adaptive transform also exists so that perfect reconstruction of the images is possible.

### 2.3. Compression of Images

A 3D mesh is represented by two image-like signals by applying the proposed mesh-to-image transform. As these signals are on a regular grid they can be compressed using any image coder. Two issues defining the mesh quality are; (i) Length of the used bitstream for SPIHT & Quantization level for JPEG2000, (ii) Number of wavelet decomposition levels. Decreasing the length of the bitstream leads to more compression at the expense of higher distortion. Increasing the number of wavelet decomposition levels usually leads to higher compression ratios at the expense of more computational cost.

The distortion level of the reconstructed 3D mesh is measured visually by *MeshTool* [10] or using some tools like *METRO* [11]. *Mean Square Error* (MSE) and “*Hausdorff Distance*” (Haus.Dist.) between the original and the reconstructed object are mostly used error measures in the literature.

The proposed SPIHT based approach is a progressive compression scheme since the leading bits of the code streams are low pass part of the signal and the latter parts are the details. So pruning the leading bits causes a higher distortion in the reconstructed model than pruning the ending bits. By first reconstructing the model using the leading bits and then refining it using the newly coming stream is possible.

The issue of how much of the SPIHT stream should be taken and what should be the quantization levels of JPEG2000 are closely related to the detail level parameter used in the orthogonal projection operation. If the detail level is low, the percentage of the used bitstream or number of quantization levels must be increased to reconstruct the 3D mesh without much distortion.

### 2.4. Reconstruction from SPIHT Bitstream

The 3D mesh is reconstructed using the SPIHT or JPEG2000 bitstream and some other side information, such as the vertex indexes (the second channel), the detail level used in the image-like representation. First the bitstream is transformed to image-like representation by decoding. The inverse of the adaptive wavelet transformation is applied to the image using the connectivity information.

Finally, using the projected vertex coordinates and the vertex indices, the image is back-projected to the 3D space. Since the only exact data available is the orthogonal component  $\hat{\mathbf{d}}_i$  of the 3D mesh vertices, the mesh cannot be perfectly reconstructed. Some vertices may coincide while they are projected onto the  $\mathbf{n}$  plane. The connectivity list is used to find the neighbors of the lost vertex. Thus, the lost vertices can be predicted from their connected neighbors.

Model	Compression Algorithm	Data Size-(KB)	Max Dist. Hausdorff	Mean Dist.
Homer	MPEG	41.8	0.002645	0.000660
Homer	SPIHT	9.41	0.003704	0.000600
Homer	SPIHT	7.92	0.005216	0.000930
Cow	MPEG	26.1	0.001780	0.000680
Cow	SPIHT	7.25	0.005631	0.000410
Lamp	MPEG	36	0.430000	0.100000
Lamp	SPIHT	4.77	0.014680	0.001700
9Torus	MPEG	82.8	0.001563	0.005976
9Torus	SPIHT	12.7	0.009797	0.000927
Sandal	MPEG	22.7	0.001904	0.000743
Sandal	SPIHT	5.91	0.007705	0.000273
Sandal	SPIHT	4.2	0.020076	0.000788
Dance	MPEG	55.4	0.002007	0.000673
Dance	SPIHT	17.3	0.003393	0.000326
Dance	SPIHT	12.1	0.009140	0.001060
Dragon	MPEG	43.1	0.001473	0.000557
Dragon	SPIHT	7.18	0.056720	0.001920

**Table 1.** Comparative results for the Homer, 9Handle Torus, Sandal, Dragon, Dance models compressed using MPEG-3DGC and SPIHT mesh coders. Hausdorff distances are measured between the original and reconstructed meshes.

## 3. RESULTS

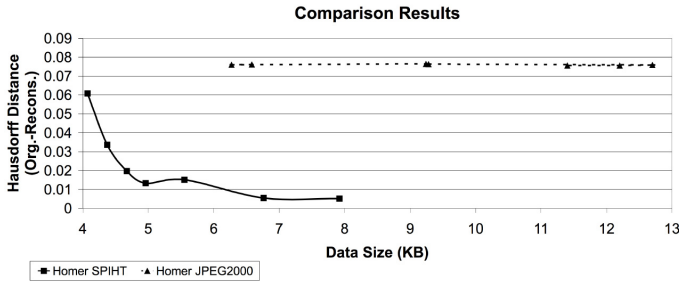
The proposed framework is tested by compressing the 9 Torus and Homer Simpsons models. The 9Handle Torus model is obtained from <http://www.ics.uci.edu/~pablo/files/data/genus-non-0/9HandleTorus.ply> and is composed of 9392 vertices with 165 KB compressed data size. The Homer Simpson model is obtained from *INRIA Gamma Team Research Database Website Collections* and is composed of 4930 vertices with 98 KB data size.

In Figure 3 the rate distortion values for compressed Homer Simpson and 9Handle Torus models are given. It can be observed from Figure 3 that meshes compressed with SPIHT are superior to JPEG2000 compressed meshes. In Table 1 a comparison between the SPIHT based method and MPEG-3DGC [12] is given. When the same *mean distance* between the original and reconstructed models taken into account, the SPIHT coder’s data size is superior to the MPEG-3DGC coder. Error in the high pass parts of the models is due to the lost vertices in the projection operator. If a better projection operator can be defined, those error can be corrected and much better results can be obtained.

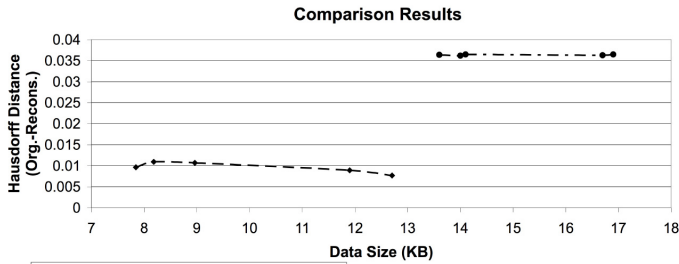
## 4. CONCLUSIONS

The results in this paper show that the idea of using image processing tools on meshes can be realized without making any parametrizations [7] on the mesh or manipulations on the used image processing tools [8]. By comparing the results in Figure 3 with results in [9], it can be said that for the same distortion level, adaptive scheme has lower bit rates than non-adaptive scheme. Thus, adaptive approach is superior to the non-adaptive approach. This is due to the better exploitation of the correlation between connected vertex positions.

In this work connectivity adaptive wavelet transform is embedded into two famous wavelet based image compressors; *SPIHT* and *JPEG2000*. As it is seen in Figure 3, the SPIHT coding is superior



(a)



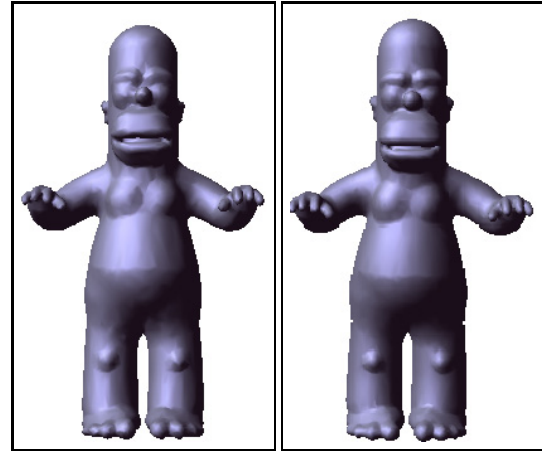
(b)

**Fig. 3.** Compression results for (a) Homer and (b) 9Handle Torus models using SPIHT and JPEG2000. Hausdorff distances are measured between the original and reconstructed models.

to the JPEG2000 coder in rate-distortion measure. It can be concluded from Table 1 that the results of the SPIHT coding is superior to MPEG-3DGC coder. For the same distortion rate (mean Hausdorff Distance), bitstream created by SPIHT coder has a lower data size.

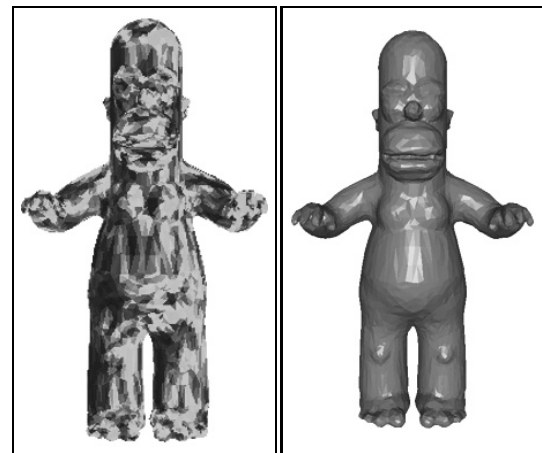
## 5. REFERENCES

- [1] P. Alliez and C. Gotsman, "Recent advances in compression of 3D meshes", In *Advances in Multiresolution for Geometric Modelling*, N.A. Dodgson, M.S. Floater and M.A. Sabin (eds), Springer-Verlag, pp. 3-26, 2005.
- [2] J. Peng, C.-S. Kim, Kuo, and C.-C. Jay, *Technologies for 3D triangular mesh compression: a survey*, Technical Report, University of Southern California, 2003.
- [3] F. Moran and N. Garcia, "Comparison of wavelet-based three-dimensional model coding techniques", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 14, No. 7, pp. 937-949, 2004.
- [4] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees, *IEEE Trans. Circ. Syst. Video Tech.*, Vol. 6, pp. 243-250, 1996.
- [5] J.M. Shapiro, "Embedded image coding using zerotrees of wavelets coefficients", *IEEE Trans. Signal Processing*, Vol. 41, pp. 3445-3462, 1993.
- [6] JPEG 2000 Part I Final Committee Draft, Version 1.0, 2000.
- [7] X. Gu, S. Gortler, H.Hoppe, "Geometry Images", *Proc. of ACM SIGGRAPH*, pp. 355-361, 2002.
- [8] I. Guskov, W. Sweldens and P. Schröder, "Multiresolution Signal Processing for Meshes", *Proc. of ACM SIGGRAPH*, pp. 325-334, 1999.
- [9] K. Köse, A. E. Çetin, U. Güdükbay, L. Onural, "Nonrectangular wavelets for multiresolution mesh analysis and compression", *Proc. of SPIE Defense and Security Symposium, Independent Component Analysis, Wavelets, Unsupervised Smart Sensors, and Neural Networks IV*, Vol. 6247, pp. 19-30, 2006.
- [10] N. Aspert, D. Santa-Cruz and T. Ebrahimi, "MESH: Measuring Error between Surfaces using the Hausdorff distance", *Proc. of the IEEE International Conference on Multimedia and Expo (ICME)*, vol. 1, pp. 705-708, 2002.
- [11] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: measuring error on simplified surfaces", *Computer Graphics Forum*, Vol. 17, No. 2, pp. 167-174, 1998.
- [12] MPEG 3D Graphics Compression group, [www.mpeg-3dgc.org](http://www.mpeg-3dgc.org).



(a)

(b)



(c)

(d)

**Fig. 4.** Homer model reconstructed from (a) 9.41 KB of SPIHT bitstream and (b) 41.8 KB of MPEG bitstream. (c) Face mean error of the Homer model reconstructed from SPIHT bitstream. Darker regions represents low error regions. (d) Flat-shaded Homer model reconstructed SPIHT bitstream.