Eurographics Symposium on Geometry Processing (2003)
L. Kobbelt, P. Schröder, H. Hoppe (Editors)

Geometry Compression of Normal Meshes Using Rate-Distortion Algorithms

Sridhar Lavu, Hyeokho Choi and Richard Baraniuk

Department of Electrical and Computer Engineering, Rice University, Houston, Texas, USA

Abstract

We propose a new rate-distortion based algorithm for compressing 3D surface geometry represented using triangular normal meshes*. We apply the Estimation-Quantization (EQ) algorithm to compress normal mesh wavelet coefficients. The EQ algorithm models the wavelet coefficients as a Gaussian random field with slowly varying standard deviation that depends on the local neighborhood and uses rate-distortion optimal scalar quantizers. We achieve gains of 0.5 to 1 dB with the EQ algorithm compared to the recently proposed zerotree compression for normal meshes.*

Categories and Subject Descriptors (according to ACM CCS): E.4 [Coding and Information Theory]: Data Compaction and Compression G.1.2 [Numerical Analysis]: Approximation - *approximation of surfaces and contours, wavelets and fractals* I.3.4 [Computer Graphics]: Computational Geometry and Object Modeling - *hierarchy and geometric transformations*

1. Introduction

Three-dimensional (3-D) surface applications require modeling complex 3-D surface geometry[18]. The models are typically represented using a mesh of polygons such as triangles or quadrilaterals. Typically complex 3-D surfaces require a large number of polygons which can result in an enormous amount of data; for example, the 3-D surface representing Michelangelo's statue of David contains about a billion triangles[18]. Efficient mesh storage and mesh transmission require the need for efficient mesh compression algorithms.

The 3-D polygon mesh data consists of two parts: the vertex data and the mesh connectivity. The vertex data specifies the location of the vertices of the polygons in the mesh and the mesh connectivity represents how the vertices are connected to form polygons. In the naive representation of the mesh, we need three numbers to represent each vertex in 3-D coordinate space and three numbers to represent the vertices of a triangle. In this paper, we focus on triangular meshes with geometry information.

A multiresolution mesh representation is obtained using *Subdivision*[1]. Depending on the structure of the mesh, meshes can be classified as regular, semi-regular, or irregular. Regular and semi-regular meshes are commonly used in 3-D modeling as they enable compaction of the connectivity information using simple schemes. In particular, semi-regular meshes offer the flexibility of starting with a simple irregular mesh at the coarsest scale, and successively regularly subdividing each triangle to obtain meshes at multiple resolutions. For irregularly spaced data points such as triangular mesh data, *lifting* combined with *subdivision* naturally defines a *wavelet transform* on the mesh[4]. Furthermore, by restricting new vertices to lie only in the normal direction specified by the local coordinate system, we can represent each vertex with just one parameter to obtain a *normal mesh*[6] representation.

Normal meshes are particularly attractive for mesh compression, because they have the ability to represent each vertex using a single parameter rather than the naive representation of using three coordinates to represent a vertex. A number of efficient multiresolution mesh compression algorithms have been proposed, of which the progressive zerotree normal mesh compression gives the best results[13]. The zerotree algorithm arranges the wavelet coefficients in a quadtree and applies the zerotree compression algorithm originally developed for still images[15]. Another recent mesh compression scheme that has been proposed for irregular

meshes uses vertex-quantization to compress the vertices of the mesh[16]. This algorithm assumes that the connectivity of the irregular mesh algorithm is compressed separately.

In this paper, we develop a new compression algorithm using rate-distortion optimized techniques that have been originally developed for still image compression. The normal mesh wavelet coefficients exhibit high spatial and inter-scale correlations. In smooth regions, the wavelet coefficients tend to be small, while in rough regions they are large. Our algorithm exploits these correlations to encode the normal mesh wavelet coefficients.

We model the normal mesh wavelet coefficients as an outcome of a random field with a slowly varying standard deviation. To compress the normal mesh wavelet coefficients using the local intra-scale correlations, we adopt the *Estimation-Quantization* (EQ) framework that has been successfully applied in still image compression[7]. The statistics of each wavelet coefficient are modeled using a local standard deviation. A rate-distortion (R-D) optimal quantizer is chosen followed by an entropy coder for coding the quantization symbols of each wavelet coefficient.

Compared to the state-of-the-art zerotree mesh compression algorithm[13], our EQ mesh coder provides 0.5-1 dB improvement in coding performance.

In Section 2, we briefly review the theory of normal triangular meshes and the zerotree mesh compression algorithm. We investigate the statistics of the normal mesh wavelet coefficients, review the EQ coder for images and describe the details of the EQ mesh coder in Section 3. In Section 4, we briefly discuss the compression of the base mesh geometry and the connectivity information. In Section 5, we demonstrate the performance of our proposed compression algorithm for typical 3-D meshes and compare it with the state-of-the-art zerotree mesh compression algorithm. Section 6 concludes the paper with suggestions for future research.

2. Previous work

2.1. Normal triangular meshes

In this paper, we will focus our attention on triangular semi-regular normal meshes[6]. In a normal mesh, the vertices are arranged in a multiscale representation where new vertices are added at each scale; each new vertex lies in the direction normal to a surface fit through the local neighborhood of vertices as shown in Figure 1. A subdivision scheme, such as the modified butterfly subdivision scheme[3], can be used to predict the normal direction and the base point for each new vertex using the coarse scale vertices in the local neighborhood of the new vertex. The intersection of the normal line with the original surface gives the new vertex. In this fashion, each coarse scale triangle is regularly subdivided to obtain the mesh at the next finer resolution. Figure 2 shows the Venus head normal mesh at the base level and Figure 3 shows the mesh after one level of subdivision along with the vertices at the previous scale.

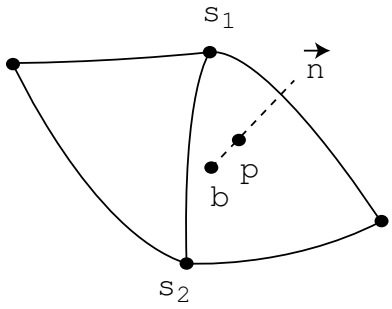


Figure 1: *The new vertex p associated with edge (s_1, s_2) is given by the intersection of the normal line $\vec{n}$ passing through base point b with the original surface.*

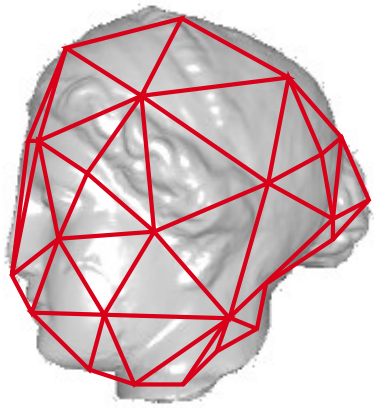


Figure 2: *The base mesh of the Venus head normal mesh.*

Lifting[4] combined with subdivision can be used to define a wavelet transform such as the butterfly wavelet transform[3] which is based on the modified butterfly subdivision scheme. The wavelet transform of a normal mesh results in a vector wavelet coefficient for each new vertex. The vector wavelet coefficients are typically expressed using the local coordinates with one normal and two tangential components for each vertex. However, since each new vertex of a normal mesh lies in a direction normal to the local surface defined using the coarse scale vertices, the wavelet coefficients of a majority of the vertices can be expressed using just the normal component of the wavelet coefficient for each new vertex.

The geometry of a smooth surface can be described using a single normal component of the wavelet coefficient for each vertex, as the tangential components of the normal mesh wavelet coefficients are either small or zero in smooth

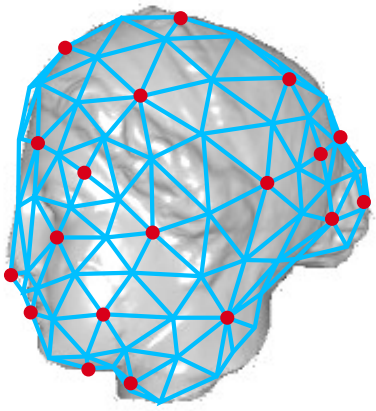


Figure 3: *The Venus head normal mesh after one level of subdivision. The thick dots are the vertices of the mesh at the previous coarser scale.*

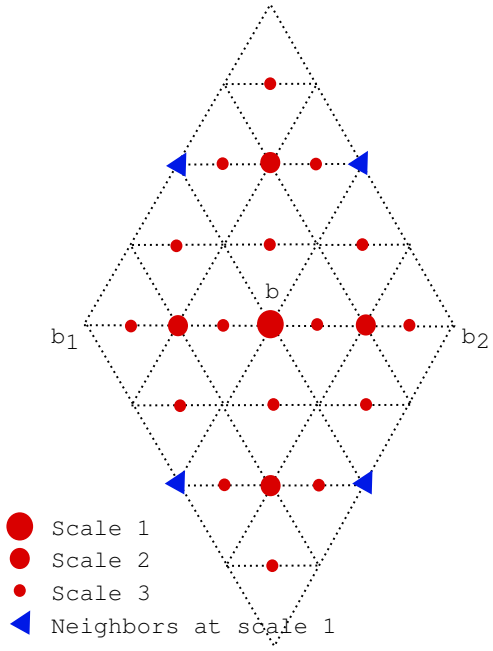


Figure 4: *Quadtree of vertices for edge b_1, b_2. The vertices of the quadtree at scales 1,2 and 3 are shown in the figure using thick dots. The neighboring vertices of vertex b at scale 1 are shown using triangles.*

regions. Thus, normal meshes offer a significant redundancy reduction over other mesh representations.

2.2. Zerotree mesh compression

The mesh wavelet coefficients can be arranged in a multiresolution quadtree structure as shown in Figure 4. The zerotree mesh coder exploits this quadtree structure and encodes each component of the mesh wavelet coefficients separately in the same way that the zerotree image coder does[15, 13].

3. EQ mesh coding

3.1. Wavelet coefficient statistics

The wavelet coefficient vector for each vertex consists of one normal and two tangential components. The distribution of the normal wavelet coefficients at an intermediate scale of the Venus head normal mesh is shown in Figure 5.

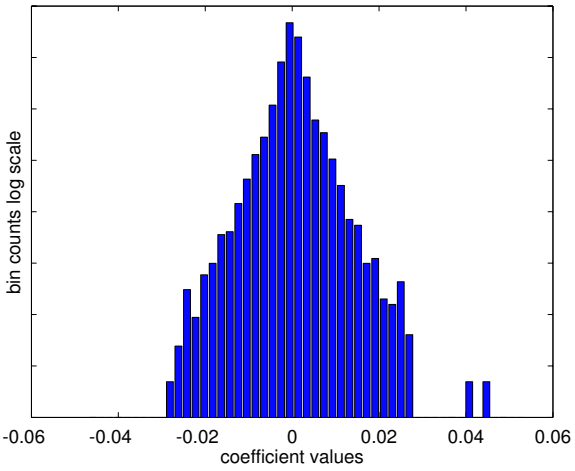


Figure 5: *Distribution of the normal wavelet coefficients of the Venus head normal mesh.*

The distribution of the normal wavelet coefficients scaled by the standard deviation of the local neighborhood of same-scale vertices is shown in Figure 6. This distribution can be approximated with a Gaussian density function as shown in the figure. Encoding the *normalized* normal wavelet coefficients gives a better performance compared to encoding the original normal wavelet coefficients as shown by the R-D curves in Figure 7. The rate and distortion measures used here are discussed in Section 5. Thus, we are motivated to model the normal wavelet coefficients using a Gaussian random field with slowly spatially varying standard deviation.

The quantized coefficients in the causal neighborhood are used to obtain the standard deviation estimate for each normal wavelet coefficient. This allows the decoder to use the same algorithm to obtain the same standard deviation estimate. However, since the the resulting probability density

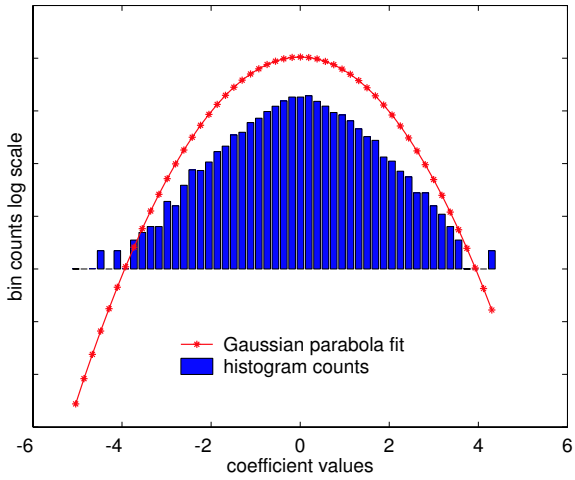


Figure 6: *Normal wavelet coefficients scaled by the standard deviation of the local neighborhood for the Venus head normal mesh.*

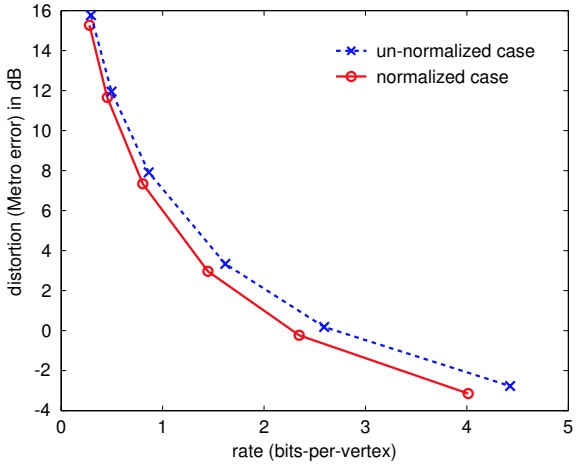


Figure 7: *R-D curve comparing the encoding of the original and scaled normal wavelet coefficients for the Venus head normal mesh.*

obtained using the quantized neighborhood is more peaky compared to a Gaussian distribution[8], we approximate the distribution with a GGD given by

$$f_{(\nu,\mu,\sigma)}(X = x) = \frac{\nu\eta(\nu,\sigma)}{2\Gamma(1/\nu)}\, e^{-[\eta(\nu,\sigma)|x-\mu|]^{\nu}} \qquad (1)$$

where $\eta(\nu,\sigma) = \frac{1}{\sigma}\sqrt{\frac{\Gamma(3/\nu)}{\Gamma(1/\nu)}}$, $0 < \nu \leq 2$, ν is the shape, μ is the mean, and σ is the standard deviation. The shape ν controls the peakiness of the distribution at each scale.

A majority of the tangential wavelet coefficients are either zero or small as shown in the distribution of the tangential wavelet coefficients from an intermediate scale of the Venus head normal mesh in Figure 8. Therefore, we model the tangential wavelet coefficients as *iid* zero-mean generalized Gaussian distribution (GGD) given by Equation (1) with fixed unknown standard deviation and shape parameters.

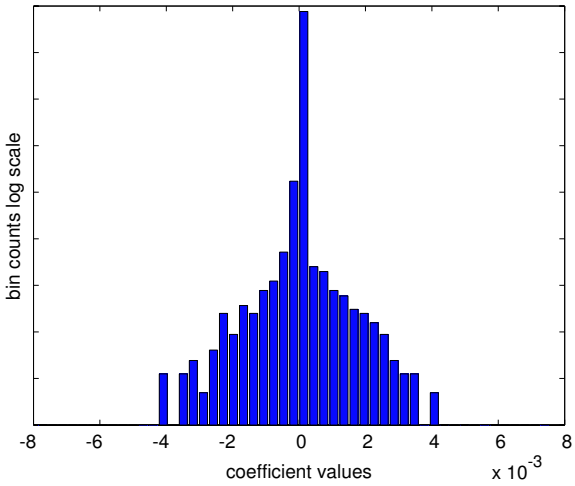


Figure 8: *Distribution of the tangential wavelet coefficients of the Venus head normal mesh, with small range on the horizontal axis.*

The EQ algorithm[7] has two main steps, the estimate step and the quantization step. The estimate step is used to estimate the shape and standard deviation parameters for the normal and tangential wavelet coefficients, and the quantization step is used to encode the coefficients using a R-D optimization. In the next section, we briefly discuss the EQ coder for still images and then introduce our EQ coder for meshes in the subsequent section.

3.2. EQ image coding

The EQ coder was originally proposed for still image compression[7]. The coefficients from each wavelet subband of the image are encoded using an independent GGD model with a slowly spatially varying unknown standard deviation and a fixed unknown shape parameter.

In the estimate step, a simple raster scan order is used to traverse through the list of the coefficients in each wavelet subband. In the simplest form of the EQ algorithm, the statistics of the GGD model for each pixel are obtained from the quantized causal subset of the local 3×3 pixel neighborhood. Figure 9 shows the causal neighborhood of a pixel and the raster scanning order of the pixels in a still image. The maximum likelihood estimate of the standard deviation for each pixel is obtained from this neighborhood[7]. In the quantize step, the estimate for the standard deviation is used to

choose an R-D optimal scalar dead-zone quantizer for each wavelet coefficient.

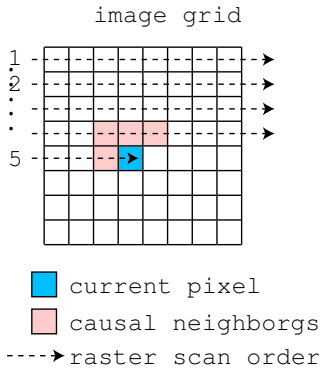


Figure 9: *Causal neighborhood and raster scanning order of pixels in a still image.*

The R-D optimization[20] is based on the Lagrangian cost function[19]. The cost function for each wavelet coefficient of pixel i is given by

$$J_i = R_i + \lambda D_i \qquad (2)$$

where λ is the Lagrangian parameter or the R-D slope, R_i is the rate expressed in terms of the entropy associated with the corresponding bin probabilities of the quantizer, and D_i is the distortion expressed in terms of the squared quantization error associated with each quantizer. Figure 10 shows the R-D curve and the R-D operating point given by the slope λ. The R-D trade off is captured in the Lagrange cost function given in Equation (2). The corresponding bin probabilities are used to encode the quantization symbol with an entropy coder such as an arithmetic coder[14]. The bin probabilities for each quantizer from the R-D table are computed off-line to speed up the algorithm.

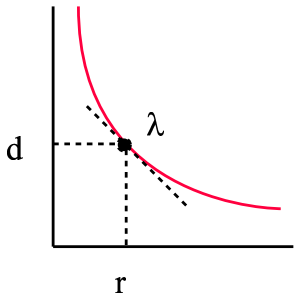


Figure 10: *Rate-distortion curve.*

This simple EQ algorithm results in a stability problem when all the causal neighbors of a coefficient are quantized to zero. In this scenario, the standard deviation estimate obtained from the causal neighbors is zero and the coefficient is automatically quantized to zero. This phenomenon can propagate through the rest of the coefficients at the same level. A simple solution to this problem is to classify the coefficients with all causal neighbors quantized to zero into a separate set called the *unpredictable set*, and place the remaining coefficients in the *predictable set*. The coefficients in the unpredictable set are modeled as *iid* zero-mean GGD with fixed shape and standard deviation. These parameters are estimated from the wavelet coefficients in the unpredictable set and sent to the decoder.

The initial partitioning of the coefficients into the predictable and the unpredictable sets at each scale determines the initial estimates for the shape parameters of the predictable and the unpredictable sets and the initial estimate for the standard deviation parameter of the unpredictable set. This initial partitioning of the wavelet coefficients into the predictable and the unpredictable sets is obtained by a simple thresholding technique[7].

3.3. EQ normal mesh coding

Our EQ mesh coder uses the local statistics of the normal wavelet coefficients to estimate the shape ν and the standard deviation σ for the each vertex as mentioned in Section 3.1.

In the estimate step, the local neighborhood of a vertex is used to estimate the standard deviation for the normal wavelet coefficient of the given vertex. Figure 11 shows a vertex v and its local neighbors at different scales. From the figure, we can see that there is no obvious choice for the vertex neighborhood; in particular, the simple neighborhood consisting of the four adjacent same-scale neighbors as shown in Figure 11 is too small to obtain a reasonable estimate of the standard deviation. Therefore, we expand the neighborhood to include the first three layers or rings of same-scale neighbors as shown in Figure 12.

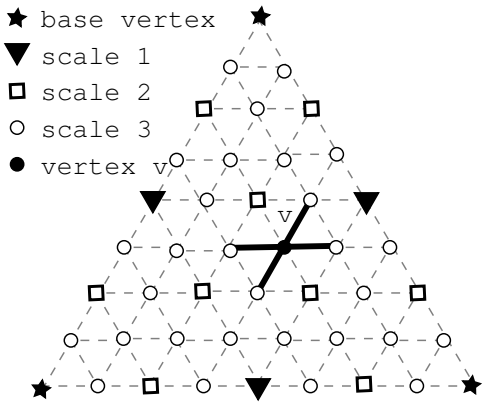


Figure 11: *A base triangle showing vertices at different scales. The four same-scale neighbors of vertex v are connected to v with solid edges.*

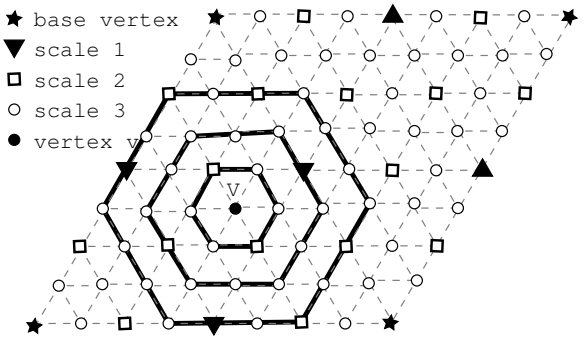


Figure 12: *Two base triangles showing the neighborhood of a vertex v. We use three layers of vertices at the same scale as v for the neighborhood. Note that for the vertex v, we include vertices from both the base triangles.*

Unlike the image case, there is no obvious choice for an ordering of the vertices. We propose a structured ordering for the vertices that improves the number of causal neighbors in the local vertex neighborhood and preserves the semi-regular connectivity of the normal mesh. In this ordering for the vertices, we start with the vertices at the coarse level and then proceed to finer levels. At each level, each base triangle is scanned separately. Thus, the global ordering of the vertices is now reduced to an ordering of the base triangles followed by an ordering of the vertices inside each base triangle.

Figure 13 shows the ordering of vertices inside each base triangle, where the ordering starts from the bottom-left and then moves right and up. The orientation of the base triangle is selected such that the number of causal same-scale neighbors is maximized for all the vertices inside the base triangle. Note that while constructing the causal vertex neighborhoods for the vertices of the current base triangle as shown in Figure 12, the vertices from the neighboring base triangles are also included in the neighborhood.

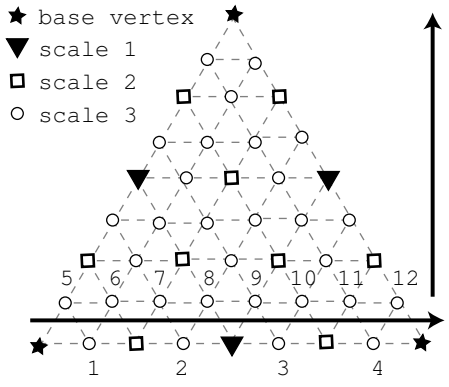


Figure 13: *Scale 3 vertex ordering inside a base triangle.*

Note that it is possible to use a more global ordering where we go through all the vertices with a single sweep of the entire mesh. However with such an ordering, it is difficult to preserve the semi-regular mesh connectivity. With our proposed ordering of the vertices, the semi-regular mesh connectivity is perfectly preserved and the decoder can easily reconstruct the mesh connectivity at the finer scales using the base mesh connectivity.

Using this local neighborhood and scanning order definitions, the standard deviation and the shape estimates are obtained for each vertex from its causal quantized neighborhood. A scalar dead-zone quantizer is selected using a R-D optimization, given by Equation (2), where the rate R_i is expressed in terms of the entropy of the bin probabilities, and the distortion D_i is measured using the mean-squared quantization error. The quantized symbols are then encoded using the corresponding bin probabilities and an entropy coder such as an arithmetic coder.

The EQ mesh coder has the same stability problem as the EQ image coder when all the quantized normal wavelet coefficients in the causal neighborhood of a vertex are zero. Therefore, we classify these coefficients as *unpredictable* in the same manner as discussed in Section 3.2.

The tangential wavelet coefficients are modeled as *iid* zero-mean GGD with fixed shape and standard deviation parameters for each scale. A single scalar dead-zone quantizer is chosen for all the vertices at each scale based on a R-D optimization. For a smooth surface, a small displacement in a direction normal to the surface changes the original surface much more than the same displacement in a direction tangential to the surface. Therefore, the tangential wavelet coefficients are encoded using a higher R-D slope λ compared to the normal components.

The algorithm is summarized in the following steps.

1. Arrange each base triangle and all the vertices inside that base triangle in a triangular array form as shown in Figure 11.
2. Find scanning order of base triangles.
3. Set scale $j = 1$.
4. If scale $j >$ depth, then skip to Step 19..
5. Set iteration index $i = 0$.
6. Partition the normal components of the wavelet coefficients into predictable and unpredictable sets using a simple thresholding technique.
7. If $i > 0$ and if the predictable and unpredictable sets have converged, then jump to Step 17..
8. For all the vertices at the current scale j, find the estimates of the standard deviation and the shape parameters for the tangential components and the unpredictable set of normal components of the wavelet coefficients.
9. Scan through the remaining vertices at the current scale that are in the predictable set of normal components of wavelet coefficients using the scanning order shown in Figure 13.

10. For each vertex at scale j, find the causal neighborhood as shown in Figure 12.
11. Using the quantized values of the normal wavelet coefficients in the causal neighborhood, find the estimate of the standard deviation for the current vertex.
12. If the estimate is zero, then classify the current vertex as unpredictable normal wavelet coefficient.
13. Using the R-D optimization, find the best scalar quantizer for the current vertex.
14. Encode the quantized value using an entropy coder.
15. The predictable and unpredictable sets of the normal components of the wavelet coefficients from this iteration are used for the next iteration.
16. If all the vertices at the current scale are encoded, then increase iteration index from i to $i+1$ and return to Step 7., else return to Step 10..
17. Increase scale from j to $j+1$.
18. Return to Step 4.
19. Encode scaling coefficients and base mesh connectivity.
20. Send GGD parameters and λ as side information to the decoder.

Further details of the EQ mesh coder can be obtained from the EQ algorithm used for still image compression[7].

4. Scaling coefficients and mesh connectivity

The vertices of the base mesh correspond to the scaling coefficients in a wavelet transform. The scaling coefficients are uniformly quantized using variable bit rate. Note that the scaling coefficients are expressed in terms of the absolute coordinate system while the wavelet coefficients are expressed as local offsets in a local coordinate system. Therefore, the scaling coefficients have a significantly higher global error contribution compared to the wavelet coefficients for the same squared error.

The proposed vertex ordering preserves the semi-regular connectivity of the normal mesh. Therefore, we only need to encode the connectivity of the base mesh. We encode the base mesh connectivity using the TG coder[11]. At the decoder, the base mesh is regularly subdivided to obtain the connectivity at the finer resolutions.

5. Results

The performance of our EQ mesh coder is analyzed using the Venus head, the horse, and the rabbit normal meshes[13].

The Metro tool[12] is used to measure the final error. The Metro tool measures the squared symmetric distance between two surfaces averaged over the first surface. We refer to this error as the RMS Metro error. PSNR † values are expressed as a ratio of the bounding box diagonal over the RMS Metro error, where the bounding box diagonal is the longest diagonal of the box that bounds the original surface. The rate is expressed in terms of the bits-per-vertex values, given by the ratio of total number of bits over the number of vertices in the original irregular mesh.

We apply the unlifted and the lifted versions of the butterfly wavelet transforms[3] to the normal mesh, since the butterfly subdivision scheme was used to construct the normal mesh from the original irregular mesh. The wavelet coefficients are encoded using our EQ algorithm. The scaling coefficients are uniformly quantized and then encoded using an entropy coder. The base mesh connectivity is encoded using a TG coder[11]. The Metro error is measured between the original irregular and the reconstructed normal meshes.

We fix the value of λ for all normal mesh coefficients in the mesh. We use a higher λ for the tangential wavelet coefficients, as their error contribution is much smaller than the normal wavelet coefficients as discussed in Section 3.3. In our simulations, we used 100λ for the tangential wavelet coefficients. The normal wavelet coefficients are encoded using our EQ mesh coder.

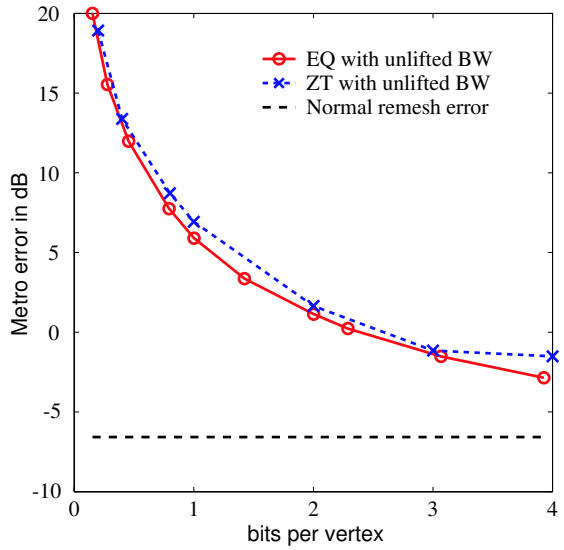
The performance of the EQ mesh coder for the Venus head normal mesh is compared with that of the state-of-the-art zerotree mesh coder[13]. Figure 14 shows the R-D curves for the EQ mesh coder and the zerotree mesh coder and compares the distortion with the normal remeshing error, which is the error between the original irregular mesh and the original normal mesh. PSNR values as a function of the bits-per-vertex are given in Table 1.

Table 1: *Venus head mesh PSNR in dB † comparing the EQ mesh coder and the zerotree (ZT) mesh coder using the lifted and unlifted versions of the butterfly wavelet (BW) transform.*

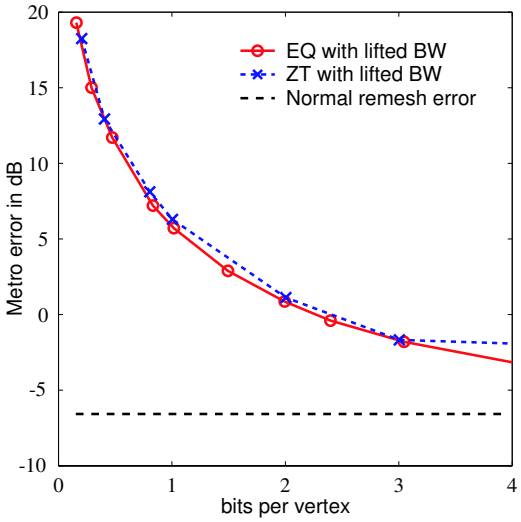
bits-per-vertex	EQ lifted BW	ZT lifted BW	EQ unlifted BW	ZT unlifted BW
0.25	63.69	62.95	63.48	62.40
0.5	68.63	68.21	68.59	67.78
1.0	74.16	73.66	74.08	73.00
2.0	79.16	78.85	78.85	78.40
3.0	81.70	81.66	81.37	81.20
4.0	83.16	81.92	82.96	81.50

The experiments are repeated for the rabbit normal mesh. Figure 15 shows the R-D curves for the EQ mesh coder and the zerotree mesh coder and compares the distortion with the normal remeshing error. PSNR values as a function of the bits-per-vertex are given in Table 2. We obtained similar results for the horse normal mesh dataset.

† PSNR in $dB = 20\log\left(\frac{\text{bounding box diagonal}}{\text{Metro error}}\right)$.

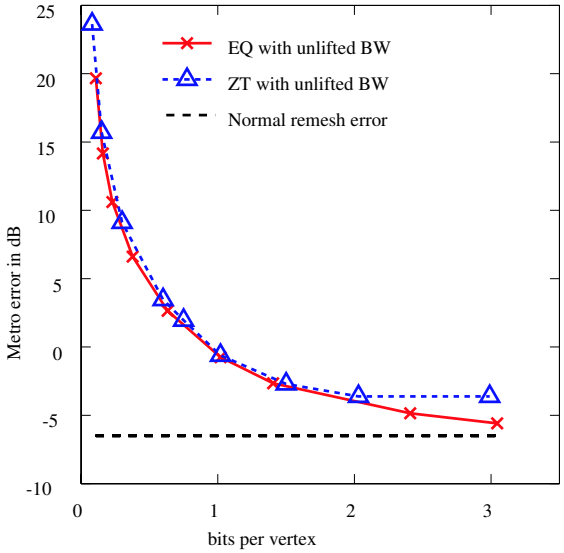


(a) *Unlifted case.*

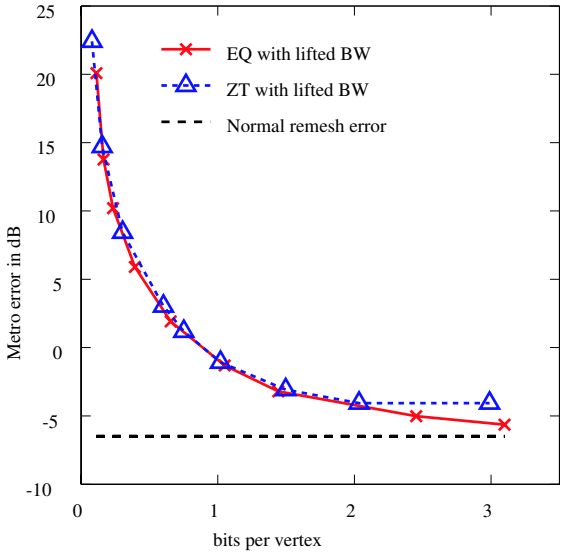


(b) *Lifted case.*

Figure 14: *R-D curves comparing the EQ mesh coder, the zerotree (ZT) mesh coder, and the normal remeshing error for the Venus head normal mesh using the unlifted and lifted versions of the butterfly wavelet transform (BW). Note that the normal remeshing error gives an approximate lower bound for the final error.*



(a) *Unlifted case.*



(b) *Lifted case.*

Figure 15: *R-D curves comparing the EQ mesh coder, the zerotree (ZT) mesh coder, and the normal remeshing error for the rabbit normal mesh using the unlifted and lifted versions of the butterfly wavelet transform (BW). Note that the normal remeshing error gives an approximate lower bound for the final error.*

Table 2: *Rabbit mesh PSNR in dB* † *comparing the EQ mesh coder and the zerotree (ZT) mesh coder using the lifted and unlifted versions of the butterfly wavelet (BW) transform.*

bits-per-vertex	EQ lifted BW	ZT lifted BW	EQ unlifted BW	ZT unlifted BW
0.16	66.24	65.29	65.84	64.29
0.37	74.10	72.82	73.38	72.20
1.02	81.31	81.07	80.75	80.61
1.4	83.21	83.07	82.65	82.27
3.0	85.64	84.06	85.59	83.61

6. Conclusions

In this work, we have presented a context-based EQ coder for normal meshes and compared it with the state-of-the-art zerotree coder for normal meshes. The EQ coder gives performance gains of approximately 0.5-1 dB. Although the EQ mesh coder is not strictly progressive like the zerotree mesh coder, it is scale-wise progressive. Since the mesh data is *smoother* compared to the image pixels, the coding of the wavelet coefficients was found to be easier in the mesh data case.

In the future, we hope to improve the results for the EQ algorithm by replacing the mean squared error (MSE) metric with a vertex-based error metric while performing the R-D optimizations, as a vertex based error metric would better correspond with the final global error measurement compared to the MSE. We hope to study the R-D trade-offs between the scaling coefficients, the normal wavelet coefficients, and the tangential wavelet coefficients.

The zerotree based Space-Frequency-Quantization (SFQ) algorithm is another R-D optimization based algorithm that was originally proposed for still images[9]. We are in the process of implementing the SFQ algorithm for normal meshes and hope to achieve better results compared to the EQ and zerotree mesh coders.

Acknowledgments

We thank Peter Schröder, Andrei Khodakovsky, and Igor Guskov from the Multi-Res Modeling group at Caltech for many interesting discussions and for providing the normal mesh data and wavelet transforms.

References

1. J. Warren and H. Weimer. *Subdivision Methods for Geometric Design: A Constructive Approach*. Morgan Kaufmann Publishers, October 2001
2. C. Loop. Smooth subdivision surfaces based on triangles. Master's Thesis, University of Utah (August 1987).
3. D. Zorin, P. Schröder, and Wim Sweldens. Interpolating Subdivision for Meshes with Arbitrary Topology. *Computer Graphics Proceedings*, **30**:189–192, August 1996.
4. W. Sweldens The Lifting Scheme: A Construction of Second Generation Wavelets. *SIAM Journal Math. Analysis*, **29**(2):511–546, 1998.
5. A. Khodakovsky, P. Schröder, and Wim Sweldens. Progressive Geometry Compression. *ACM Computer Graphics (Proceedings of SIGGRAPH '00)*, 271–278, 2000.
6. I. Guskov, K. Vidimce, W. Sweldens, and P. Schröder. Normal Meshes. *ACM Computer Graphics (Proceedings of SIGGRAPH '00)*, 95–102, 2000.
7. S. M. Lopresto, K. Ramchandran, and M. T. Orchard. Image Coding based on Mixture Modeling of Wavelet Coefficients and a Fast Estimation-Quantization Framework. *Proceedings DCC'97 (IEEE Data Compression Conference)*, 221–230, 1997.
8. S. M. Lopresto, K. Ramchandran, and M. T. Orchard. Wavelet image coding via rate-distortion optimized adaptive classification. *Proceedings of NJIT Symposium on Wavelet, Subband and Block Transforms in Communications*, New Jersey Institute of Technology, 1997.
9. Z. Xiong, K. Ramchandran, and M. T. Orchard. Space-Frequency Quantization for Wavelet Image Coding. *IEEE Transactions Image Processing*, **6**(5):677–693, May 1997.
10. A. Lewis and G. Knowles. Image compression using the 2-D wavelet transform. *IEEE Transactions Image Processing*, **1**:244–250, April 1992.
11. C. Touma and C. Gotsman. Triangle Mesh Compression. *Graphics Interface*, 26–34, June 1998.
12. P. Cignoni, C. Rocchini, and R. Scopigno. Metro: measuring error on simplified surfaces. *Computer Graphics Forum*, **17**(2): 167–174, June 1998.
13. A. Khodakovsky and I. Guskov. Normal mesh compression. *submitted for publication*, `http://www.multires.caltech.edu/pubs/compression.pdf`.
14. Ian H. Witten, Radford M. Neal, and John G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, **30**(6):520–540, 1987.
15. J. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions Signal Processing*, **41**:3445-3462, December 1993.

16. P. Chou and T. Meng. Vertex Data Compression through Vector Quantization. *IEEE Transactions Visualization and Computer Graphics*, **8**(4):373–383, 2002.

17. T. DeRose, M. Kass, and T. Truong. Subdivision surfaces in character animation. *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 85–94, 1998.

18. M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The Digital Michelangelo Project: 3D Scanning of Large Statues. *ACM Computer Graphics (Proceedings of SIGGRAPH '00)*,131–144, 2000.

19. D. A. Wismer and R. Chattergy. Introduction to Nonlinear Optimization. *Elsevier North Holland Publishers*, 1978.

20. R. M. Gray. Source Coding Theory. *Kluwer Academic Press*, 1990.