3-D Mesh Geometry Compression with Set Partitioning in the Spectral Domain

Ulug Bayazit, Umut Konur, and Hasan Fehmi Ates, *Member, IEEE*

***Abstract*—This paper explains the development of a highly efficient progressive 3-D mesh geometry coder based on the region adaptive transform in the spectral mesh compression method. A hierarchical set partitioning technique, originally proposed for the efficient compression of wavelet transform coefficients in high performance wavelet based image coding methods, is proposed for the efficient compression of the coefficients of this transform. Experiments confirm that the proposed coder employing such a region adaptive transform has a high compression performance rarely achieved by other state of the art 3-D mesh geometry compression algorithms.**

A new, high performance fixed spectral basis method is also proposed for reducing the computational complexity of the transform. Many-to-one mappings are employed to relate the coded irregular mesh region to a regular mesh whose basis is used. To prevent loss of compression performance due to the low-pass nature of such mappings, transitions are made from transform based coding to spatial coding on a per region basis at high coding rates. Experimental results show the performance advantage of the newly proposed fixed spectral basis method over the original fixed spectral basis method in the literature that employs one-to-one mappings.

***Index Terms*—Data compression, data visualization, transform coding, computer graphics, virtual reality.**

I. Introduction

In recent years, there has been a marked increase in the demand to access and visualize 3-D object information in a wide variety of applications in manufacturing, architecture, defense, computer aided design and entertainment. Most applications that employ 3-D models are network-bound. Consequently, conservation of bandwidth by compressing the geometry and topology of the surface model has been in the focus of research efforts since the seminal work of Deering [1] and gained speed with the development of the MPEG-4 Visual. Functionality and low complexity as well as compression efficiency became the key elements of the more recent 3-D object surface representation and coding algorithms.

Three types of information are coded for the surface mesh representation of an object. *Connectivity* information specifies all pairs of connected vertices. *Geometry* information provides the 3-D coordinates of all vertices. Additionally, properties associated with each face, edge or vertex of an object such as texture values and material attributes might also be coded.

Most early geometry compression methods are predictive schemes that are tightly coupled with the connectivity coding methods. The vertex traversal order dictated by connectivity coding is accepted by both the encoder and decoder in such coders. A wide range of predictive geometry coders [1]–[9] for single or multi-resolution, lossless or lossy coding based on scalar or vector quantization have been previously proposed. Exceptions to the rule of connectivity coding dictating geometry coding may be found in [10], [11].

Since the advent of [13], [14], transform domain mesh geometry coding has gained acceptance. Most wavelet transform based geometry compression methods [14]–[17] treat the compression problem as a 3-D object surface approximation problem which is different from the irregular mesh vertex approximation problem treated in [13], [18]–[20], [28]. The methods in [14]–[17] convert irregular mesh connectivity into subdivision connectivity by a remeshing technique [9] to yield a multi-level representation. An iterative subdivision process is carried out on a base level coarse mesh to obtain finer level meshes at the desired resolutions. In normal mesh compression [15], a base point is predicted from vertices in the coarse level and a finer level vertex is determined by intersecting a line drawn in a normal direction from the base point with the original surface. Thereby, most of the finer level vertices are expressed by a single (normal) coordinate in a local coordinate system. Advanced coding algorithms such as SPIHT (Set Partitioning in Hierarchical Trees, [21]) or Estimation/Quantization [16] are then used to compress the normal and tangential coordinates.

In the progressive compression scheme of [28], the wavelet decomposition is formulized for an irregular mesh without altering the original connectivity. The wavelet transform is realized by an exact integer analysis-synthesis scheme via the lazy filter-bank modified by the lifting scheme. Starting with a base mesh, connectivity is iteratively coded by coding for each face, the positions of the newly inserted vertices, subdivision orientation and edge flips for some subdivisions into two. The detail information (wavelet coefficients) of each level is scalar quantized and entropy coded. In [29], a zerotree based bitplane coder is substituted for the scalar quantizer of [28].

The transform of [13], the focus of the current work, is a generalization of the classical 1-D Fourier transform to the 3-D irregular meshes. The transform coefficients are adaptively computed by projecting the vertex coordinates onto the orthonormal Laplacian basis derived from mesh topology. In the spectral compression method of [13], most of the coefficients with low presumed energy are discarded, and the rest are entropy coded. This coder is embedded to some extent because

U.B. is with Istanbul Tech. Univ., 34469, Maslak, Istanbul, TURKEY (phone:+90 212 285 3591; e-mail: ulugbayazit@itu.edu.tr)

U.K.. , Bogazici Univ., 34342 Bebek, Istanbul, TURKEY (e-mail: konur@boun.edu.tr)

H.A. Isik Univ., Kumbaba Mevkii, 34980, Sile, Istanbul, TURKEY (e-mail: hfates@isikun.edu.tr.)

Manuscript received xxxxxxx, 2008; revised xxxxxxx,2009.

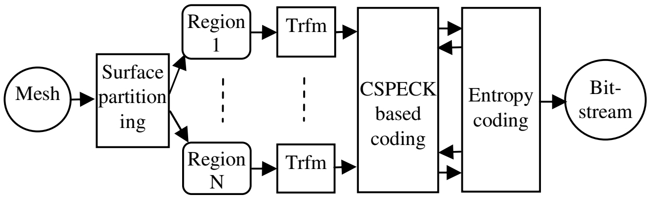

Fig. 1. Proposed CSPECK based geometry encoder

increasing the number of coded/decoded coefficients allows for progressive reconstructions at increasing fidelity levels. The extensions in [18]–[20] attempt to reduce the transform complexity or to improve the visual quality.

The current work solely addresses geometry coding since recent research has largely focused on it and the coordinate values contribute the greater fraction of the 3-D mesh information. The entire connectivity information is assumed to be coded in advance as in [13], [28], [29] and facilitates the single resolution, lossy coding of the geometry information by the proposed coder. The proposed coder primarily supports triangular faces that are the common denominator of rendering tools in graphics hardware and software.

The geometry coder proposed in this paper is depicted in Figure 1. Surface partitioning limits the complexity of the transform of [13] for a large surface mesh and facilitates the local surface properties to be better captured for compression purposes since the transform for each region is adaptively designed. As in [13], the MeTiS software package [23] is also used here for multilevel graph based mesh partitioning.

We first propose that the transform coefficients for all the regions be bit plane coded with the CSPECK (Color Set Partitioning Embedded Block Coder) algorithm [22] with a motivation similar to that of [14], [15], [29]. The CSPECK algoritm is based on SPECK, the low complexity, high performance wavelet based greyscale image compression algorithm that was integrated into the JPEG 2000 standard as SBHP (Subband Block Hierarchical Partitioning). CSPECK processes the coefficients in an order dictated by their magnitudes using a set partitioning method and makes implicit bit allocation to each space coordinate or mesh region possible. The more accurate bit priorities yield a truly embedded bitstream. In addition, since the dependencies among the most significant bits of multiple coefficient magnitudes can be exploited by coding multiple insignificant bits with a single symbol, high compression performance can be attained.

A second important contribution of this work is a modification of the fixed spectral basis method of [18] toward a low complexity and high performance coding system. Although the coding methods in [13]–[15] provide high compression performance, their front end components such as the adaptive transform and the semi-regular remesher are computationally inefficient. Due to the resulting inferior compression performance, the use of the original fixed spectral basis method of [18] in place of the adaptive transform is not a good solution to the complexity problem. The compression performance suffers not because of the basis itself, but because of the way the irregular mesh region vertices are one-to-one mapped to the regular region vertices prior to the transform. As a viable solution, we propose that the coordinate values of more than one irregular mesh region vertex be used to define the coordinate values of each regular region vertex at the encoder, and the coordinate values of more than one reconstructed regular region vertex be used to define the coordinate values of each reconstructed irregular mesh region vertex at the decoder.

Next section reviews the adaptive transform and the spectral mesh compression method of [13]. Section III describes the proposed bit plane coding of the transform coefficients, illustrates the basic operation on a simple example mesh and elaborates on the entropy coding of the generated symbols. Section IV reviews the fixed spectral basis method of [18] and presents the proposed fixed basis method based on many-to-one mappings between the irregular region vertices and the regular region vertices. Section V presents the experimental results for the proposed coder with either the adaptive transform or the proposed fixed spectral basis method, as well as for the methods of [3], [13], [18], [28] for comparison.

II. The Spectral Mesh Compression Method

In [13], the transform for coordinate $\gamma \in \{x, y, z\}$ is performed by projecting s_γ, the vector of γ coordinate values for all vertices in a region, onto the orthonormal basis defined by the eigenvectors of the Laplacian operator matrix L with

$$L_{ij} = \begin{cases} 1 & i = j \\ -1/d_i & i \text{ and } j \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases}$$

as elements where d_i denotes the valence of the i^{th} vertex of the region. The Laplacian operator transforms the coordinate values into their prediction error values.

For regions with largely smooth surfaces, such a basis is advantageous for compression purposes since it compacts most of the prediction error information into a small number of low frequency coefficients corresponding to small eigenvalues. Figure 2 shows the energy distribution of the coefficients (sorted by their eigenvalues from small to large) of the mostly smooth regions of the Bunny model. Prediction error information in regions of models with high frequency detail such as sharp edges, corners and crevices can be represented with a large number of coefficients.

Spectral mesh compression method in [13] keeps a subset of the coefficients corresponding to small eigenvalues, scalar quantizes these coefficients and entropy codes the quantization levels. High rate-distortion performance is not guaranteed due to inefficient bit allocation to coefficients. For example, a low bit plane bit of a small eigenvalue coefficient with low energy may be coded prior to a high bit plane bit of a large eigenvalue coefficient with high energy.

A practical problem with the spectral mesh compression method is the size of the alphabet for coding the quantization levels at high quantization resolutions. A resolution of 16 bits requires an alphabet of 32768 letters with 1 bit reserved for the coefficient sign. In adaptive entropy coding, this implies that a very large number of symbols ($>> 32768$) need to be coded before the probability model approximates the actual

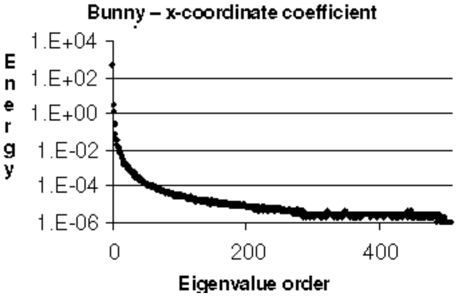

Fig. 2. A sharp decline in the energies of coefficients (sorted in the ascending order of their eigenvalues) is observed.

probability mass distribution of the levels. The long warm-up period implies a substantial increase of rate over entropy.

III. Bit-plane coding with CSPECK

The SPECK algorithm [22] recursively partitions sets of wavelet coefficients to locate the large magnitude coefficients and code their most significant bits with priority in the bitstream. Unlike its predecessor SPIHT, which exploits the correlation between the magnitudes of spatially related coefficients both across and within subbands, SPECK exploits the correlation between the magnitudes of spatially related coefficients only within a subband for image compression gain.

In the color image coding method of [22], the SPECK operations for the 2-D wavelet transform coefficients of the color planes are interleaved in each pass. Similarly, we adopt CSPECK for coding 1-D coefficient vectors of arbitrary size by interleaving the SPECK operations for the three 1-D coefficient vectors of the three coordinates in each pass.

A. CSPECK for coding 1-D coefficient vectors

Prior to the application of CSPECK, the spectral transform is applied on the vertices of each region to compute the coefficients. For each coordinate $\gamma \in \{x, y, z\}$, the coefficients are ordered by their eigenvalues (small to large) to make up the components of a coefficient vector c_γ of arbitrary size. Such an ordering that generally arranges the coefficients from high to low energy, is the configuration with the largest correlation between adjacent coefficient magnitudes that does not need to be specified to the decoder with overhead bits.

Both SPECK and CSPECK consist of multiple coding passes. When SPECK that operates on a single set is extended to CSPECK for coding the three 1-D coefficient vectors for the three space coordinates, the (sub)sets belonging to different vectors are processed largely independently of each other, in an interleaved manner.

Let $t_{\gamma,i}$ denote the i^{th} component of a vector (set) t_γ of coefficients for coordinate γ. The significance of the vector at bitplane n is defined as

$$\Pi_n(t_\gamma) = \begin{cases} 1 & 2^n \leq \max_i\{|t_{\gamma,i}|\} < 2^{n+1} \\ 0 & \text{otherwise} \end{cases}$$

Let the most significant bitplane n_{max} be such that $\Pi_{n_{max}}(c_\gamma) = 1$ for at least one γ and $\Pi_{n_{max}+1}(c_\gamma) = 0$ for all γ. As CSPECK is initialized, n_{max} is coded. In $(n_{max} - n + 1)^{th}$ pass bitplane n is processed. In the first pass, each of the three coefficient vectors is partitioned into a *type S* set S_0 and a *type I* set I_0 as shown in Figure 3. In each following pass, CSPECK recursively partitions significant sets down to significant coefficients or smaller insignificant sets. When partitioned, a generic type I set I_r yields a smaller type I set I_{r+1} and a type S set S_{r+1} so that $I_r = I_{r+1} \cup S_{r+1}$. In this case, $|S_0| = 2^l$ for some small integer l and $|S_r| = \min(\sum_{k=0}^{r-1} |S_k|, |I_{r-1}|)$ with $I_{-1} = c_\gamma$. When a generic type S set S of arbitrary size is partitioned, two type S sets, S' and S'' are formed where $|S'| = \lfloor |S| /2 \rfloor$ and $|S'| = \lceil |S| /2 \rceil$.

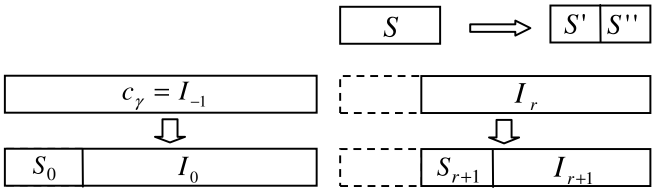

Fig. 3. Left: Initial partitioning of the coefficient vector. Right: Generic partitioning schemes for sets of type S (top) and type I (bottom).

The location information for a significant coefficient or an insignificant set gets implicitly coded as a consequence of coding the set significance decisions.

CSPECK maintains, LIS, a list of type S sets deemed insignificant for the most recent bitplane, and LSC, a list of coefficients deemed significant for the most recent bitplane.

In the first step of each pass called the sorting step, the type S sets in the LIS are recursively processed. This is followed by the recursive processing of the type I sets (if any). A type S set is processed by coding its significance state and partitioning it into two type S sets as described above when it is deemed significant and contains more than one coefficient. The type S sets in the LIS are processed in the order of increasing cardinality since fewer bits are coded for small type S sets for about the same distortion reduction as large S sets. Due to recursion in processing, all significant coefficients in these sets are encountered in the same sorting step. A type I set is processed by coding its significance state and partitioning it into a type S set and a smaller type I set as described above when it is deemed significant. Insignificant type S or type I sets are added to the LIS. Finally, the signs of significant coefficients are coded in the sorting step. These coefficients are added to the LSC by recording their first reconstructions (representing the most significant bits) as $\hat{c}_{\gamma,i} = \pm(1.5)2^n$ and the reconstruction errors as $e_{\gamma,i} = c_{\gamma,i} - \hat{c}_{\gamma,i}$.

In the refinement step, the less significant bits of those coefficients deemed significant before the last sorting step are output as $b_{\gamma,i} = (e_{\gamma,i} > 0)$. The reconstructions and the errors are updated as $\hat{c}_{\gamma,i} = \hat{c}_{\gamma,i} + (2b_{\gamma,i} - 1)2^{n-1}$ and $e_{\gamma,i} = c_{\gamma,i} - \hat{c}_{\gamma,i}$.

Finally, in the *quantization step*, the bitplane index n is updated as $n \leftarrow n - 1$. The three steps of each pass are iteratively executed until the desired rate is reached.

B. A simple example

In this section, some of the critical details of the coding procedure are highlighted on a sample triangle mesh with 4 vertices and 4 triangular faces. For the sample mesh, $G = \begin{bmatrix} -3 & 2 & 2 \\ 7 & -1 & 9 \\ 2 & 8 & 7 \\ 8 & 3 & 5 \end{bmatrix}$ and $N = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$ (ij^{th} entry indicates whether vertices i and j are connected) are the geometry and adjacency matrices, respectively. The corresponding Laplacian operator matrix is $L = \begin{bmatrix} 1.000 & -0.333 & -0.333 & -0.333 \\ -0.333 & 1.000 & -0.333 & -0.333 \\ -0.333 & -0.333 & 1.000 & -0.333 \\ -0.333 & -0.333 & -0.333 & 1.000 \end{bmatrix}$ whose column eigenvectors are stacked into the matrix $\Psi = \begin{bmatrix} -0.500 & -0.216 & 0.866 & 0.089 \\ -0.500 & 0.839 & -0.289 & -0.345 \\ -0.500 & -0.146 & -0.289 & -0.520 \\ -0.500 & -0.478 & -0.289 & 0.776 \end{bmatrix}$ in the order of increasing eigenvalues which are $0, 1.333, 1.333$ and 1.333.

The transform coefficients for each coordinate, computed by projecting the geometry onto the eigenvectors, make up the rows of the matrix

$$C = G^T\Psi = \begin{bmatrix} -7.000 & 2.408 & -7.506 & 2.491 \\ -6.000 & -3.868 & -1.155 & -1.311 \\ -11.500 & 3.712 & -4.330 & -2.685 \end{bmatrix}.$$

Since the maximum coefficient magnitude is 11.500, we have $n_{max} = \lfloor \log_2 11.500 \rfloor = 3$. The proposed coder works in a normalized domain for magnitudes where the maximum magnitude is 1.4375 so that $2^{n_{max}} \times 1.4375 = 11.5$.

In this example, each c_γ is initially partitioned into a single coefficient type S set S_0 and a type I set I_0 of three coefficients. When I_0 is significant, it is partitioned into another single coefficient type S set S_1 and a type I set I_1 of two coefficients. When I_1 is significant, it is partitioned into two single coefficient type S sets S_2 and S_3. When S_i is significant its sign is coded and it is added to the LSC.

Even though actual coding with CSPECK takes place until the desired bit rate is reached, let us examine only the first five passes in our example. For each pass, the bitplane index n, the magnitude threshold 2^n in that pass, and the significance states, the coded signs and the reconstructions of all coefficient magnitudes after that pass are shown in Table I.

The coded refinement decisions at each step (not shown in Table I) can be inferred. For example, in pass 1, no refinement code is produced whereas in pass 2, those coefficients that were significant in pass 1 are refined. Observing the reconstructed magnitudes of the coefficients at the end of each pass reveals what the refinement decisions are.

At the end of the fifth pass, the matrix of coefficient reconstructions is $\hat{C} = \begin{bmatrix} -7.25 & 2.25 & -7.75 & 2.25 \\ -6.25 & -3.75 & -1.25 & -1.25 \\ -11.75 & 3.75 & -4.25 & -2.75 \end{bmatrix}$. The geometry matrix of reconstructed coordinates is then determined as

TABLE I
THE BITPLANE INDEX n, THE MAGNITUDE THRESHOLD T_n , THE SIGNIFICANCE STATES OF THE COEFFICIENTS AND SIGNS OF SIGNIFICANT COEFFICIENTS AND THE RECONSTRUCTIONS OF THE COEFFICIENTS AT EACH PASS.

Pass 1 $n = 3$ Threshold =8	Significance State/ Sign	x	0	0	0	0
		y	0	0	0	0
		z	1 /-	0	0	0
	Reconst. Magnitudes	x	0	0	0	0
		y	0	0	0	0
		z	12	0	0	0
Pass 2 $n = 2$ Threshold =4	Significance State/ Sign	x	1 /-	0	1 /-	0
		y	1 /-	0	0	0
		z	1 /-	0	1 /-	0
	Reconst. Magnitudes	x	6	0	6	0
		y	6	0	0	0
		z	10	0	6	0
Pass 3 $n = 1$ Threshold =2	Significance State/ Sign	x	1 /-	1 /+	1 /-	1 /+
		y	1 /-	1 /-	0	0
		z	1 /-	1 /+	1 /-	1 /-
	Reconst. Magnitudes	x	7	3	7	3
		y	7	3	0	0
		z	11	3	5	3
Pass 4 $n = 0$ Threshold =1	Significance State/ Sign	x	1 /-	1 /+	1 /-	1 /+
		y	1 /-	1 /-	1 /-	1 /-
		z	1 /-	1 /+	1 /-	1 /-
	Reconst. Magnitudes	x	7.5	2.5	7.5	2.5
		y	6.5	3.5	1.5	1.5
		z	11.5	3.5	4.5	2.5
Pass 5 $n = -1$ Threshold =0.5	Significance State/ Sign	x	1 /-	1 /+	1 /-	1 /+
		y	1 /-	1 /-	1 /-	1 /-
		z	1 /-	1 /+	1 /-	1 /-
	Reconst. Magnitudes	x	7.25	2.25	7.75	2.25
		y	6.25	3.75	1.25	1.25
		z	11.75	3.75	4.25	2.75

$$\hat{G} = \Psi^{-T}\hat{C}^T = \begin{bmatrix} -3.0818 & 2.0455 & 2.2004 \\ 6.9332 & -0.7341 & 9.1449 \\ 2.5827 & 7.9998 & 7.1164 \\ 8.0660 & 3.1889 & 5.0383 \end{bmatrix}$$

C. Lossless Coding of Symbols

Since the sign distribution of the coefficients is expected to be uniform, the sign symbols of the significant coefficients are binary-coded. Due to the high probability of the insignificance decision, the significance decision symbols are entropy coded to realize a coding gain. Similarly, as in [26], a modest coding gain is realized by entropy coding the refinement symbols since the refinement towards the small magnitude has slightly larger probability than the refinement towards large magnitude. Since the symbol alphabets are small and skewed, arithmetic coding [25] is preferred. The symbol letter probabilities are adaptively estimated from the observed frequencies during the actual coding pass as in [25].

Conditional entropy coding is used to exploit the dependencies between the decision symbols for adjacent sets. In *simple conditional entropy coding*, the significance decision symbols for two adjacent sets are individually entropy coded.

If these sets are subsets of a partitioned set, the second symbol is entropy coded only if the first one is significant since a partitioned set has at least one significant subset.

In order to further improve the coding performance, a *context adaptive conditional entropy coding* method for significance and refinement symbols has also been developed. Let Z be the context of the coded significance decision or refinement symbol X. When the coefficients are ordered by their eigenvalues, the context Z that minimizes the conditional entropy $H(X|Z) = H(X) - I(X;Z)$ (or maximizes $I(X;Z)$) is from the immediate neighborhood of X. The energy distribution in Figure 2 shows the statistical dependency between the magnitudes of adjacent coefficients.

For each space coordinate, separate adaptive probability models are trained and used in the arithmetic coding of significance and refinement symbols. Thereby, the different surface characteristics along the three coordinate axes can be captured. For instance, a region on the leg of the horse model has smaller curvature along the z-axis than along the x or y-axis. Hence, the energy distribution of the z-coordinate coefficients is more skewed towards small eigenvalues than the energy distributions of the x or y-coordinate coefficients.

Different probability models are used for coding the significance of single coefficient sets and sets with multiple coefficients. When the significance decision of a single coefficient set is coded, the (combination of the) significance state(s) of the (two) single coefficient neighbor set(s) is used as the context. Similarly, when the significance decision of a set with multiple coefficients is coded, the (combination of the) significance state(s) of the (two) neighbor set(s) of size 32 coefficients is used as the context. This empirically determined size maximizes the average coding performance.

Let T_R and T_L be the right and left neighbor sets of set T whose significance decision is coded. The significance state of T_a $(a \in \{R, L\})$ is defined as

$$\Phi_n^s(T_a) = \begin{cases} 0 & 2^n > \max\limits_{i:i\in T_a}\{|c_{\gamma,i}|\} \\ & \text{(i.e insignificant in this pass)} \\ 1 & 2^n \le \max\limits_{i:i\in T_a}\{|c_{\gamma,i}|\} < 2^{n+1} \\ & \text{(i.e significant in this pass)} \\ 2 & \max\limits_{i:i\in T_a}\{|c_{\gamma,i}|\} < 2^{n+1}, 2^n \overset{?}{>} \max\limits_{i:i\in T_a}\{|c_{\gamma,i}|\} \\ & \text{(i.e insignificant in previous pass)} \end{cases}$$

Let $p_\gamma^1(\Pi_n(T)|z)$ and $p_\gamma^m(\Pi_n(T)|z)$ be the probability models used in arithmetic coding for the single coefficient sets and multiple coefficient sets, respectively. If both neighbor sets exist, $z = 3 \times \Phi_n^s(T_L) + \Phi_n^s(T_R)$, else if only T_R exists $z = \Phi_n^s(T_R) + 9$, else if only T_L exists $z = \Phi_n^s(T_L) + 12$.

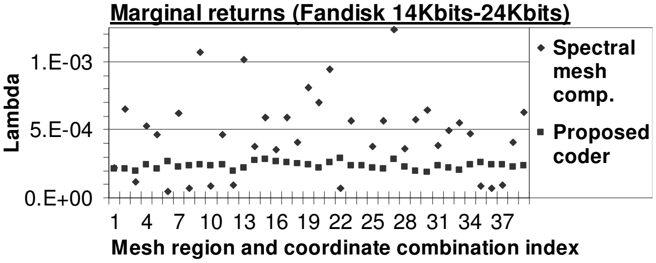

Fig. 4. The marginal returns with the proposed CSPECK based coder have much lower variability (variance=6.39E-10) than those with the spectral mesh compression method (variance=1.35E-7).

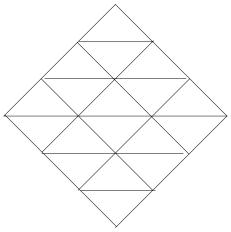

Fig. 5. A small regular hexagonal mesh with 16 vertices.

The context used for refinement symbol coding is determined using the refinement state(s) of the (two) neighbor coefficient(s) of the refined coefficient. Let $p_\gamma^r(b|z)$ be the probability model used for the arithmetic coding of the refinement symbol b of coefficient $c_{\gamma,i}$. In this case,

$$z = \begin{cases} 0 & 2^{n+1} > |c_{\gamma,i}| \\ 1 & 2^{n+1} \le |c_{\gamma,i}| \text{ and } (2^{n+1} \le |c_{\gamma,L}| \text{ or } 2^{n+1} \le |c_{\gamma,R}|) \\ 2 & 2^{n+1} \le |c_{\gamma,i}| \text{ and } 2^{n+1} > |c_{\gamma,L}| \text{ and } 2^{n+1} > |c_{\gamma,R}| \end{cases}$$

is used as the context where $c_{\gamma,R}$ and $c_{\gamma,L}$ are the right and left neighbor coefficients of the refined coefficient.

D. Rate allocation efficiency of the CSPECK based coder

The marginal return between the k'th and $k-1$'th operating rate-distortion points of n'th source is $\lambda(n,k) = \left|\frac{D_{n,k} - D_{n,k-1}}{R_{n,k} - R_{n,k-1}}\right|$. For optimal bit allocation to independently coded sources, the distortion-rate tradeoffs at operating points must be approximately equal in the sense that operating point $m(n)$ for n'th source satisfies $\lambda(r, m(r) - l) > \lambda(n, m(n))$ and $\lambda(r, m(r) + l) < \lambda(n, m(n))$ for $r \neq n$ and $l > 0$ where $R_{n,m-1} < R_{n,m}$. In the current work, each coordinate vector of each region constitutes a source. In general, the marginal returns for coding the regions with the proposed coder exhibit a much smaller variability than those for coding the regions with the spectral mesh compression method.

In Figure 4, the marginal returns for the coding of the Fandisk model with the proposed coder and the spectral mesh compression method between 14000-24000bits are presented.

IV. Fixed Spectral Basis

The fixed spectral basis method, formulated in [18] by computing the spectral basis for regular regions beforehand and using them to code the regions of an irregular mesh, offers a low complexity alternative to the adaptive transform. Figure 5 shows an example of a small hexagonal (regular) region with $N \times N = 16$ vertices.

In [18], the transforms for such regular regions of various sizes (i.e.) are designed offline as described in Section II. The smallest regular region with at least as many internal and boundary vertices as the coded irregular mesh region is

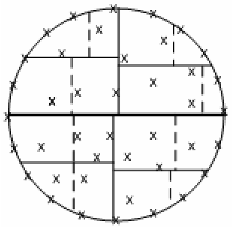

Fig. 6. Partitioning of the points (x-marks) on the unit disc.

selected. Vertices are then inserted to the irregular mesh region to make the number of its internal vertices and the number of its boundary vertices equal to those of the regular region.

A one-to-one mapping between the vertices of the coded irregular mesh region and the vertices of the regular region is established next. A geometric 2-D embedding method is proposed in [18] for determining this mapping. In this method, the boundary vertices of the regular region are represented as points on the unit circle with equal angular spacing. The internal points of the regular region are randomly initialized on the unit disc and an iterative procedure that sets each 2-D coordinate of each point equal to the mean of the corresponding coordinates of its connected neighbors is run to determine the final point positions on the unit disc. The point positions on the unit disc that represent the vertices of the irregular mesh region are determined by a similar procedure.

The points on one disc are one-to-one mapped to the points on the other disc to minimize the total squared distance. Since the geometric mapping of a large number of points between two unit discs is computationally expensive, a divide and conquer approach is followed. The points of each disc are recursively partitioned into two equal size sets by alternating horizontal and vertical lines as shown in Figure 6. The points of each smallest subset of one disc are one-to-one mapped to the points in the corresponding subset of the other disc.

In [18], the one-to-one mapping between the points in the two corresponding subsets is designed by a greedy assignment technique. A better design alternative is to use the Hungarian method, the best solution to the assignment problem.

Prior to the transform, the components of s_γ are permuted by the one-to-one mapping $M(.)$ to yield the vector $\tilde{s}_\gamma$ with components $\tilde{s}_{\gamma,j} = s_{\gamma,M(j)}$. By projecting $\tilde{s}_\gamma$ onto the spectral basis of the selected regular region, the coefficient vector c_γ is obtained for the γ coordinate which may be coded with the spectral mesh compression method of [13] or with the proposed CSPECK based coder. After decoding, the inverse transform on the reconstructed coefficient vector $\hat{c}_\gamma$ yields the vector $\hat{\tilde{s}}_\gamma$. The components of $\hat{\tilde{s}}_\gamma$ are permuted to yield the reconstructed coordinate vector with components $\hat{s}_{\gamma,j} = \hat{\tilde{s}}_{\gamma,M^{-1}(j)}$.

The inserted vertices and their reconstructions are neglected in the distortion computation, but the presence of these vertices inside s_γ has a modest contribution to the rate.

A. The proposed fixed spectral basis method

Even when the total squared distance between the points of the corresponding subsets of the unit discs is minimized, the one-to-one mapping is suboptimal. This can be illustrated with a simple example. Let point p representing vertex v in the regular region have about equal Euclidean distance to points q_1 and q_2 (with distance to q_2 slightly larger) representing vertices u_1 and u_2, respectively, in the irregular mesh region. In this case, p is one-to-one mapped to q_1, but q_2 does not take part in determining the coordinate values of v on which the transform operates. It would be more reasonable here to determine the coordinate values of v by averaging the coordinate values of u_1 and u_2 rather than using the coordinate values of only u_1.

The many-to-one mapping suggested above can be generalized. Let point p_j represent vertex $\tilde{s}_j$ in the regular region. Let points $q_{M(j,1)}, \ldots, q_{M(j,K)}$, representing vertices in the irregular mesh region, be the K nearest to the point p_j with Euclidean distances $\delta_{j,k} = \left\| p_j - q_{M(j,k)} \right\|_2$. Here $M(j,k)$ is the proposed many-to-one mapping applied before the transform. Normalization of the inverse distances yields a set of weights given as $w_{j,l} = \left(\delta_{j,l} \sum_{k=1}^{K} \delta_{j,k}^{-1} \right)^{-1}$. The j'th component of the vector of γ coordinates of the regular region vertices is computed as the weighted average of the γ coordinates of the mapped vertices in the irregular mesh region,

$$\tilde{s}_{\gamma,j} = \sum_{k=1}^{K} w_{j,k} s_{\gamma,M(j,k)}. \tag{1}$$

Figure 7 shows the proposed mapping between the two discs.

After the transform, CSPECK encoding/decoding and the inverse transform, the reconstructed coordinates of the regular region vertices are used to reconstruct the irregular mesh region coordinates by means of another similar mapping.

Let point q_j represent vertex s_j in the irregular mesh region. Let points $p_{\tilde{M}(j,1)}, \ldots, p_{\tilde{M}(j,K)}$, representing vertices in the regular region, be the K nearest to the point p_j with Euclidean distances $\epsilon_{j,k} = \left\| q_j - p_{\tilde{M}(j,k)} \right\|_2$. Here $\tilde{M}(j,k)$ is the proposed many-to-one mapping applied after the inverse transform. Normalization of the inverse distances yields a second set of weights given as $\tilde{w}_{j,l} = \left(\epsilon_{j,l} \sum_{k=1}^{K} \epsilon_{j,k}^{-1} \right)^{-1}$. The j'th component of the vector of reconstructed γ coordinates of the vertices in the irregular mesh region is computed as the weighted average of the reconstructed γ coordinates of the mapped vertices in the regular region,

$$\hat{s}_{\gamma,j} = \sum_{k=1}^{K} \tilde{w}_{j,k} \hat{\tilde{s}}_{\gamma,\tilde{M}(j,k)}. \tag{2}$$

In order to map the regular region and irregular mesh region boundaries to each other, when j is the index of a boundary vertex of the regular (irregular mesh) region, K is set to 2 and $M(j,k)$ ($\tilde{M}(j,k)$) is forced to be the index of a boundary vertex of the irregular mesh (regular) region. For many-to-one mapping of internal vertices, the coding performance increases with K, but gains are rather marginal for $K > 5$.

Similar to the original fixed basis method, the transforms for regular regions of various sizes (i.e. $N = 5, \ldots, 30$) are designed offline in the proposed method. For each irregular mesh region, the smallest regular region with more vertices is

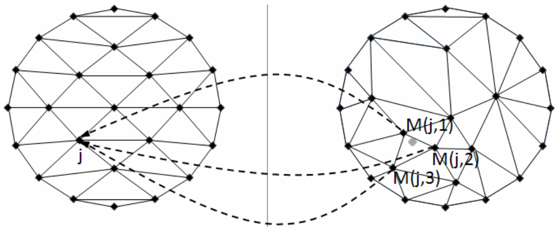

Fig. 7. The dashed lines indicate the many-to-one mapping ($N = 5, K = 3$) of the points representing the irregular region to the points representing the regular region. Position of point p_j on the left is lightly marked on the right. The solid lines indicate connected neighbor vertices

used. Insertion of vertices into the irregular mesh region is not needed since the mappings are not one-to-one.

The weighted averagings in Equations 1 and 2 are low-pass filtering type operations that result in the loss of high frequency information such as edges, corners, crevices. The CSPECK based coder employing the adaptive transform codes low frequency information at low bit rates and high frequency details at higher rates for an overall high performance. On the other hand, while the CSPECK based coder with the proposed fixed basis method also has a high performance at low rates due to coding of low frequency information, it has inferior performance at high rates due to continued coding of low frequency instead of filtered out high frequency information.

As a solution to this problem, a transition is made from transform domain coding of coordinate values to spatial domain coding of coordinate errors for each region when it is predicted in a CSPECK pass that the distortion gain for that pass and subsequent passes will be below a certain level.

Since the magnitudes of the coefficients are known prior to CSPECK coding, the reconstructed vertex coordinates in both the regular and irregular regions can be determined. If the distortion reduction $\Delta D_k = D_k - D_{k+1}$ between passes k and $k+1$ satisfies $\Delta D_k < 0.3 D_k$, $k = N, N+1, \dots$, the transform based coding of an irregular mesh region is terminated due to inadequate distortion gain for passes $N+1$ and larger.

Let $d_{desired}$ be a user specified fidelity criterion. After CSPECK based coding is completed, the coordinate errors in each irregular mesh region with $D_N > d_{desired}$ are coded in the spatial domain by successive approximation scalar quantization. The threshold δ used in the first spatial domain coding pass is set equal to half the largest vertex coordinate error magnitude which is binary coded with 32 bits. The threshold value at each subsequent pass is set equal to half the one in the previous pass. All coordinate errors are initially deemed insignificant. At each generic pass, the significance decisions for previously insignificant errors are entropy coded. The first reconstructions of significant coordinate errors are made as $\pm 1.5\delta$ by adding $\pm 1.5\delta$ to the coordinate reconstructions. At each pass, reconstructions of the coordinate errors deemed significant in previous passes are refined. According to the sign of the binary coded remaining error, $\pm 0.5\delta$ is added to the reconstructed coordinate value. The number of spatial domain coding passes needed to reach $d_{desired}$ is also binary coded by using 6 bits for each region.

In the CSPECK coder employing the fixed basis methods, the irregular mesh region is required to be simply connected since the mapped regular region is simply connected. A single application of MeTiS sometimes generates irregular mesh regions that are not simply connected. In this work, such mesh regions are recursively partitioned into two by MeTiS until all resulting smaller mesh regions are simply connected.

V. Experiments and Discussion

The simulations were conducted on the *Venus Head* (50002 vertices, 100 partitions), the low resolution *Venus Head* (8268 vertices, 16 partitions), the *Bunny* (34835 vertices, 70 partitions), the *Horse* (19,851 vertices, 40 partitions), and the *Fandisk* (6475 vertices, 13 partitions) models.

The visual error metric of [13] is preferred (see [20]) for assessing the vertex coordinate reconstruction quality, since it is based on similarities between perceptive features like local smoothness values at corresponding surface points. Let V and $\hat{V}$ be the sets of N vertices of the original and reconstructed models, M and $\hat{M}$, respectively. The metric is defined as $d_{vis}(M, \hat{M}) = \frac{1}{2N}\left(\left\|V - \hat{V}\right\|_2 + \left\|GL(V) - GL(\hat{V})\right\|_2\right)$ where $GL(v_i) = v_i - \left(\sum_{j \in n(i)} l_{ij}^{-1}\right)^{-1} \sum_{j \in n(i)} l_{ij}^{-1} v_j$ is the geometric Laplacian operator for vertex v_i , $\left\|V - \hat{V}\right\|_2 = \left(\sum_{v_i \in V} \|v_i - \hat{v}_i\|_2^2\right)^{1/2}$ and $n(i)$ is the index set of the neighbors of v_i.

Hausdorff and Root Mean Square (RMS) distances are the proper measures for assessing the approximation quality of the original mesh surface with the reconstructed surface. These distances are popular for evaluating geometry compression performance under topology change that occurs due to remeshing or simplification. The RMS distance between the surfaces S and S' is defined as $d_{rms}(S, S') = \max\left(\left(\frac{1}{|S|}\int_{p \in S} d(p, S')^2 dp\right)^{\frac{1}{2}}, \left(\frac{1}{|S'|}\int_{p \in S'} d(p, S)^2 dp\right)^{\frac{1}{2}}\right)$ where $d(p, S') = \min\limits_{p' \in S'} \|p - p'\|$ is the point to surface distance.

In Figure 8, the performance curves for two different variations of the proposed CSPECK based coder employing the adaptive transform, 12, 14 and 16 bit quantization variations of the spectral mesh compression method of [13], and the predictive compression method of [3] are presented. The spectral mesh compression method is superior to the predictive compression method at low rates, but suffers at high rates due to inefficient bit allocation to regions and coordinates. The proposed coder maintains the superior low rate performance and is competitive with the predictive compression method at high rates due to better bit allocation and the joint coding of the zero bits of multiple coefficients with a single symbol.

The context adaptive conditional entropy coding method yields a slight gain of 0.2-0.4 bpv at high rates over the simple conditional coding method. Unlike wavelet based image coding, the correlations between adjacent coefficients are not strong enough for the coder to largely exploit when the coefficients are sorted in the order of their eigenvalues.

The RMS error vs. total no. bits curves for the proposed coder and the different versions of the wavemesh coder [28],

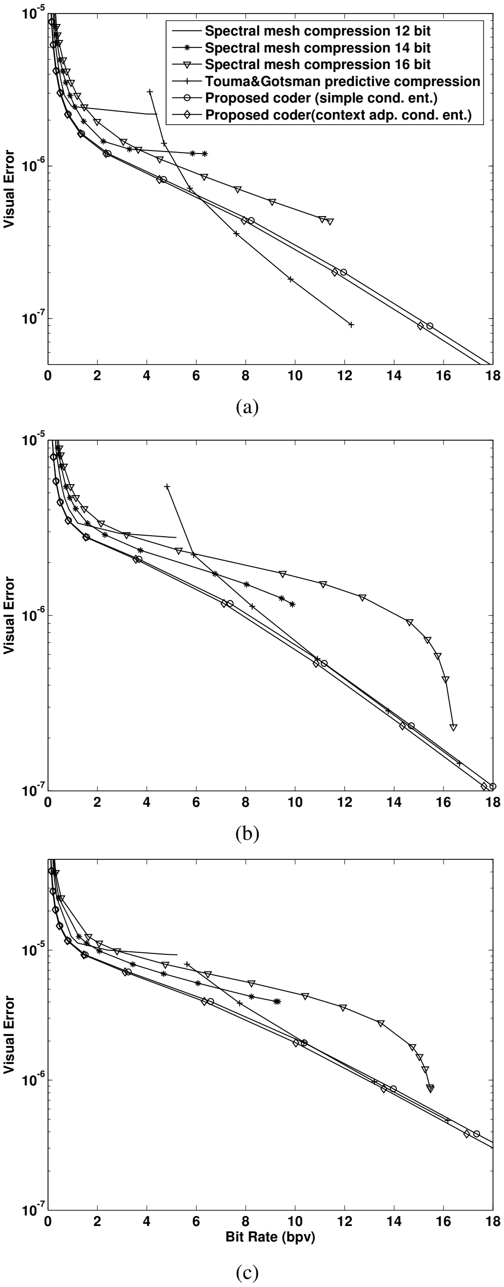

Fig. 8. The visual error vs. geometry coding rate curves for the proposed coder with the adaptive transform, the spectral mesh compression method and the predictive compression method of [3]. a) Bunny b) Horse c) Venus Head

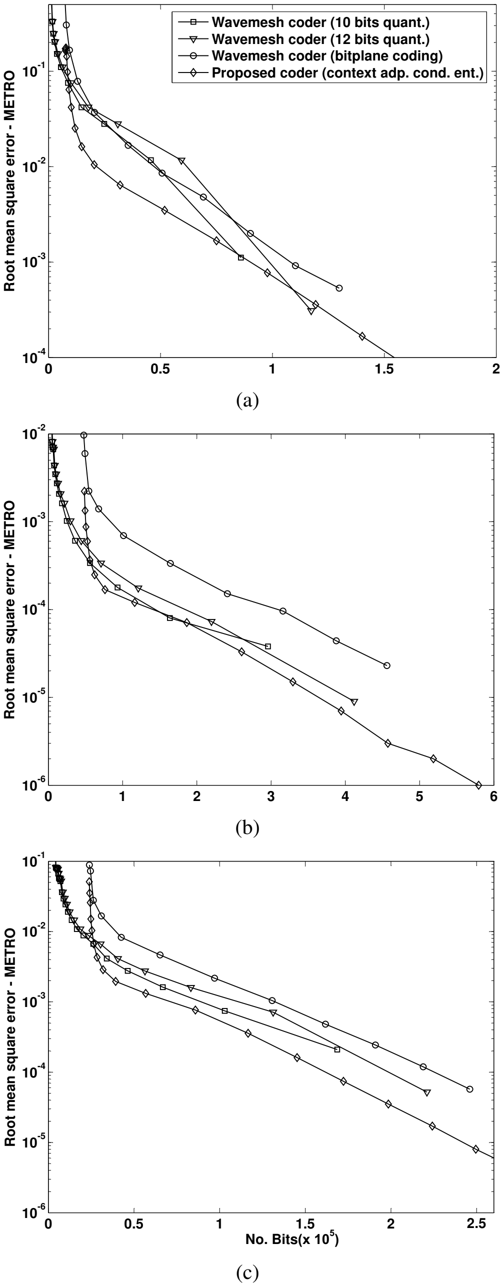

Fig. 9. The RMS vs. total no. bits curves for the proposed coder using the adaptive transform and the wavemesh coder. a) Fandisk b) Horse c) Venus Head

[29] are plotted in Figure 9. The proposed coder is superior to the wavemesh coder with the bitplane coding option at all rates and the wavemesh coder with the scalar quantization option at rates above the connectivity coding rate.

The reconstructions of Bunny and Fandisk in Figure 10 and Figure 11 suggest that the proposed coder codes high frequency features such as the edges with high priority, whereas the wavemesh coder codes the smooth regions with high priority.

In Figure 12, the performance gap between the original fixed basis method and the adaptive transform is observed to be more with the spectral mesh compression method than with the CSPECK based coder. The CSPECK based coder is more robust against the high frequency energy content of the spectrum being large with the original fixed basis method. The proposed fixed basis method partially closes this gap.

Finally, we verify the complexity advantage of the CSPECK based codec with the proposed fixed basis. Runtime measurements that exclude the time for I/O operations were taken on an Intel Core2 T7200 system with 2 GB memory and Windows XP Media Center Edn. SP 2. For the CSPECK based coder with the adaptive transform employing context adaptive conditional entropy coding, the non-optimized C-code implementations code and decode Venus Head (50002 vertices) at 16.3bits/sec. in an average of 922sec. and 1037sec, respectively. For the CSPECK based coder with the fixed basis,

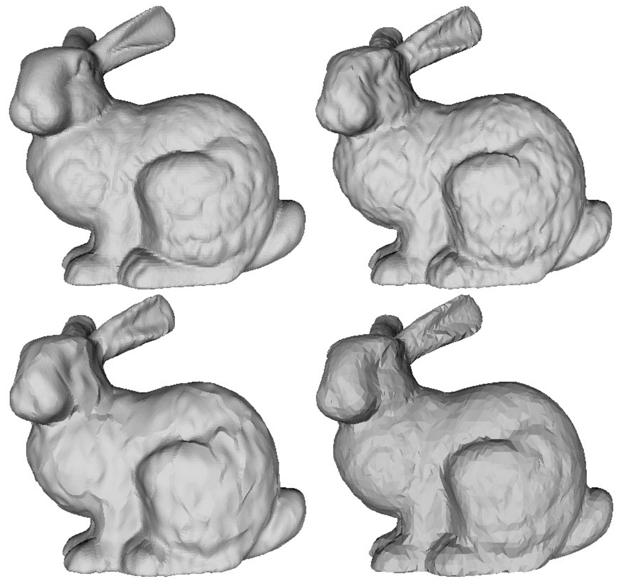

Fig. 10. Original Bunny (top left). Reconstructions: The proposed coder with the adp. transform and context adp. cond. entropy coding [top right (RMSE=1.58E-4, total 90703 bits)], Spectral mesh compression at 14 bits quant. [bottom left (RMSE=2.15E-4, total 90942 bits)], Wavemesh coder of [28] at 10 bits quant.[bottom right (RMSE=1.97E-4, total 119760 bits)]

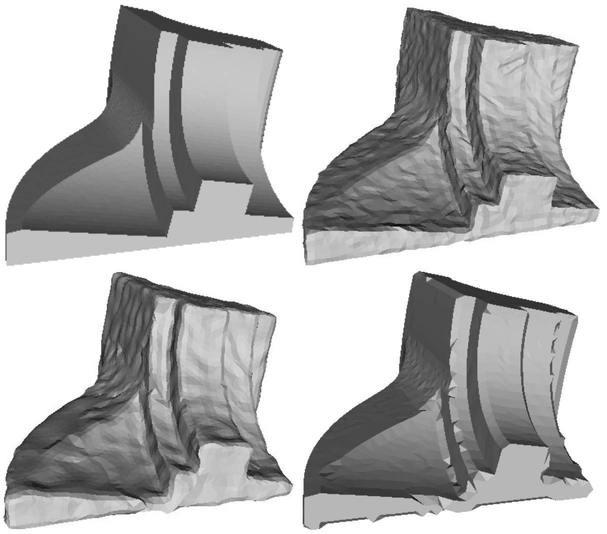

Fig. 11. Original Fandisk (top left). Reconstructions: The proposed coder with the adp. transform and context adp. cond. entropy coding [top right (RMSE=6.4E-3, 31984 total bits)], Spectral mesh compression at 14 bits quant. [bottom left (RMSE=11.2E-3, 31911 total bits)], Wavemesh coder of [28] at 10 bits quant. (note the large faults along the edges) [bottom right (RMSE=11.7E-3, 45736 total bits)]

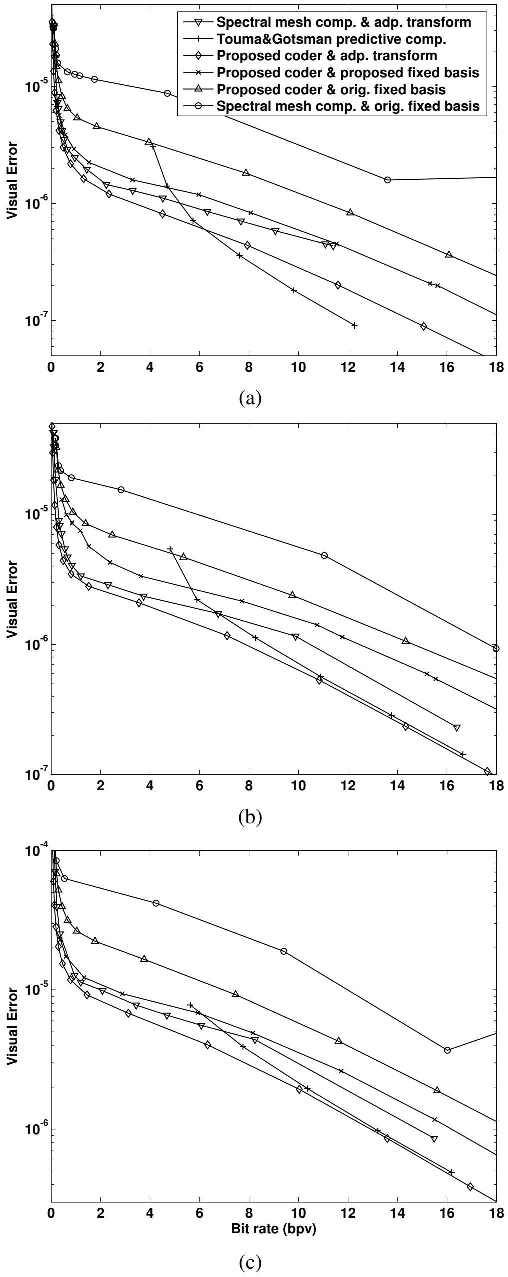

Fig. 12. The visual error vs. geometry coding rate curves for the proposed CSPECK based coder with the original fixed basis, the proposed fixed basis and the adaptive transform. The curves for the spectral mesh compression method (convex hull of the 12,14, 16 bits quantization points) with the original fixed basis and the adaptive transform, and the predictive compression method are also shown. a) Bunny b) Horse c) Venus Head

the corresponding averages are 34.8sec. and 16.8sec.

VI. Conclusion

The distortion vs. rate performance of the proposed CSPECK based coder with the adaptive transform is substantially better than that of the original spectral compression method at all geometry coding rates and generally surpasses that of the wavemesh coder at rates above the connectivity coding rate.

Replacing the adaptive transform with a fixed spectral basis reduces the computational complexity considerably. However, when integrated with the powerful CSPECK based coder, the visual error for the original fixed spectral basis method is three times that for the adaptive transform. The newly proposed spectral basis method effectively closes this performance gap.

In the future, we wish to exploit the efficient rate allocation property of the proposed coders further by pursuing the problem of joint coding of multiple models with a single CSPECK based coder. The coder is expected to allocate the available bits efficiently among different models as well as regions of each model. How close the performance with such an implicit bit allocation can approach the performance with independent coding of each model with explicit rate-distortion based bit allocation is an issue of interest.

The substitution of the diffusion wavelet basis of [24] in place of the Laplacian basis will also be investigated.

REFERENCES

[1] M. Deering, ”Geometry Compression”, *Proc. ACM SIGGRAPH*, 1995, pp. 13-20.

[2] G. Taubin and J. Rossignac, ”Geometric Compression through Topological Surgery”, *ACM Trans. Graph.* vol. 17 no. 2, pp. 84-115, April 1998.

[3] C. Touma and C. Gotsman, ”Triangle Mesh Compression”, *Proc. Graph. Interface, 1998*, pp. 26-34.

[4] M. Isenburg and P. Alliez, ”Compressing Polygon Mesh Geometry with Parallelogram Prediction,” *Proc. IEEE Visualization, 2002*, pp. 141-46.

[5] H. Hoppe, ”Progressive Meshes”, *Proc. ACM SIGGRAPH*, 1996, pp. 99-108.

[6] G. Taubin, A. Gueziec, W. Horn and F. Lazarus, ”Progressive Forest Split Compression”, *Proc. ACM SIGGRAPH*, 1998, pp. 123-132.

[7] R. Pajarola and J. Rossignac, ”Compressed Progressive Meshes”, IEEE Trans. Vis. Comp. Graph. vol. 6 no. 1, pp. 79-93, Jan.-Mar. 2000.

[8] U. Bayazit, O. Orcay, U. Konur and F. S. Gurgen , ”Predictive Vector Quantization of 3-D Polygonal Mesh Geometry by Representation of Vertices in Local Coordinate Systems,” *J. Vis. Comm. Image Rep.* vol. 18 no. 4, pp. 341-353, August 2007.

[9] E. Lee and H. Ko, ”Vertex Data Compression for Triangular Meshes”, *Proc. Pacific Graphics*, 225-234, 2000.

[10] P. M. Gandoin and O. Devillers, ”Progressive lossless compression of arbitrary simplical complexes,” *ACM Trans. Graphics.*, vol. 21 no. 3, pp. 372-379, 2002.

[11] J. Peng and C.-C. J. Kuo, ”Geometry-guided Progressive Lossless 3D Mesh Coding with Octree (OT) Decomposition,” *ACM Trans. Graphics*, vol. 24, no. 3, pp. 609-616, 2005.

[12] A. W. F. Lee, W. Sweldens, P. Schroder, L. Cowsar, and D. Dobkin, ”Maps: Multiresolution adaptive parameterization of surfaces,” *Proc. ACM SIGGRAPH*, 1998, pp. 95-104.

[13] Z. Karni and C. Gotsman, ”Spectral Compression of Mesh Geometry”, Proc. ACM SIGGRAPH, 2000, pp. 279-286.

[14] A. Khodakovsky, P. Schroder and W. Sweldens, ”Progressive Geometry Compression”, *Proc. ACM SIGGRAPH*, 2000, pp. 271-278.

[15] A. Khodakovsky and I. Guskov, ”Normal Mesh Compression,” *Geom. Modelling for Scientific Vis.*, Springer-Verlag, Germany, 2002.

[16] S. Lavu, H. Choi and R. G. Baraniuk, ”Geometry Compression of Normal Meshes using the Estimation Quantization Algorithm, *Proc. ACM SIGGRAPH/Eurograph. Symp. Geometry Proc.*, 2003, pp. 52-61.

[17] F. Payan and M. Antonini, ”An Efficient Bit Allocation for Compressing Normal Meshes with an Error-Driven Quantization’, *Comp. Aided Geom. Des. Spec. Issue Geom. Mesh Proc.* vol. 22 no. 5, pp. 466-486, 2005.

[18] Z. Karni and C. Gotsman, ”3D Mesh Compression using Fixed Spectral Bases”, *Proc. Graphics Interface Conf.*, 2001, pp. 1-8.

[19] P. Rondao-Alface, B. Macq, F. Cayre, F. Schmitt and H. Maitre, ”Lapped Spectral Decomposition for 3-D Triangle Mesh Compression”, *Proc. Int. Conf. Image Proc.*, 2003, vol. 1, pp. 781-784.

[20] O. Sorkine, D. Cohen-Or and S. Toledo, ”High-pass Quantization for Mesh Encoding”, *Proc. Eurograph./ACM SIGGRAPH Symp. Geometry Process.*, 2003, pp. 42-51.

[21] A. Said and W. A. Pearlman., ”A New, Fast and Efficient Image Codec based on Set Partitioning in Hierarchical Trees”, *IEEE Trans. Cir. Syst. Video Tech.*,vol. 6, no. 3, pp. 243-250, 1996.

[22] W. A. Pearlman, A. Islam, N. Nagaraj and A. Said, ”Efficient, Low-Complexity Image Coding with a Set-Partitioning Embedded Block Coder”, *IEEE Trans. Cir. Syst. Video Tech.* vol. 14, no.11, pp. 1219-35, 2004.

[23] G. Karypis and V. Kumar, ”MeTiS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-reducing Orderings of Sparse Matrices”, Ver. 4.0, Univ. Minnesota, Dept. Comp. Sc., http://www-users.cs.umn.edu/ karypis/metis, 1998.

[24] S. Mahadevan, ”Adaptive mesh compression in 3D computer graphics using multiscale manifold learning”, *ACM Int. Conf. Mach. Learning*, 2007, pp. 585-592.

[25] I. H. Witten, R. M. Neal and J. G. Cleary, ”Arithmetic Coding for Data Compression”, *Comm. ACM*, vol. 30, no. 6, pp. 520-540, 1987.

[26] U. Bayazit, ”Significance Map Pruning and Other Enhancements to SPIHT Image Coding Algorithm,” *Signal Proc.: Image Comm.*, vol. 18, no. 11, pp. 769-785, 2003.

[27] P. Cignoni, C. Rocchini and R. Scopigno, ”Metro: measuring error on simplified surfaces,” *Comp. Graph. Forum*, vol. 17 no. 2, pp. 167-174, 1998.

[28] S. Valette and R. Prost, ”Wavelet-Based Progressive Compression Scheme for Triangle Meshes: Wavemesh,” *IEEE Trans. Visualization and Comp. Graphics*, vol. 10, no. 2, March/April 2004.

[29] S. Valette, A. Gouaillard and R. Prost, ”Compression of 3D triangular meshes with progressive precision,” *Computers & Graphics*, vol. 28, pp. 35-42, 2004.

PLACE
PHOTO
HERE

U. Bayazit received his B.S. degree in electrical engineering from Bogazici University, Istanbul, Turkey in 1991, and his M.S. and Ph.D. degrees in electrical engineering from Rensselaer Polytechnic Institute, Troy NY in 1993 and 1996, respectively.

Dr. Bayazit worked as a research scientist and systems architect between 1996 and 2000 in Toshiba America and joined the electronics engineering department of Isik University, Istanbul, Turkey in 2000 as an assistant professor. Since 2007, he is an associate professor in the computer engineering department of Istanbul Technical University, Turkey. His research interests are in data compression and image, video, and computer graphics coding. He was the best student paper award recipient of VCIP ’97.

PLACE
PHOTO
HERE

U. Konur received his B.S. and M.S. degrees in computer engineering from Bogazici University, Istanbul, Turkey in 1999 and 2003, respectively. He is currently pursuing a Ph. D. degree at the computer engineering department of Bogazici University where he is also a research assistant. His research interests are in data compression and computer graphics.

PLACE
PHOTO
HERE

H.F.Ates (S’96-M’04) received the B.S. degree in Electrical and Electronics Engineering from Bilkent University, Ankara, Turkey, in 1998, and the M.A. and Ph.D. degrees from the Department of Electrical Engineering, Princeton University, Princeton, NJ, in 2000 and 2004, respectively. He was a Post-doc Research Associate at Sabanci University, Istanbul, between 2004 and 2005.

Dr. Ates is currently an assistant professor in the Department of Electronics Engineering, Isik University, Istanbul, which he joined in August 2005. His research interests include image, video and graphics compression, video enhancement, wavelets and multiresolution representations, and computer vision. He is currently working on industrial- and government-sponsored projects related to video coding, high-definition TV and 3-D mesh compression. He is the author/co-author of more than 20 peer-reviewed publications. He serves in the technical committee of various IEEE conferences and journals, and has been session chair at ICIP’06.