26

Optimized Scalable Wavelet-Based Codec Designs for Semi-Regular 3D Meshes

Shahid M. Satti, Leon Denis, Ruxandra Florea,
Jan Cornelis, Peter Schelkens and Adrian Munteanu
Department of Electronics and Informatics (ETRO)
Vrije Universiteit Brussel-IBBT, Brussels,
Belgium

1. Introduction

3D graphics applications make use of polygonal 3D meshes for object's shape representation. The recent introduction of high-performance laser scanners and fast microcomputer systems gave rise to high-definition graphics applications. In such applications, objects with complex textures are represented using dense 3D meshes which consist of hundreds of thousands of vertices. Due to their enormous data size, such highly-detailed 3D meshes are rather intricate to store, costly to transmit via bandwidth-limited transmission media, and hard to display on end-user terminals with diverse display capabilities. Scalable compression, wherein the source representation can be adapted to the users' requests, available bandwidth and computational capabilities, is thus of paramount importance in order to make efficient use of the available resources to process, store and transmit high-resolution meshes.

State-of-the-art scalable mesh compression systems can be divided into two main categories. A first category includes codecs that directly compress the irregular topology meshes in the spatial domain. In such codecs, the connectivity information is encoded losslessly while mesh simplification methods such as vertex coalescing (Rossignac & Borrel, 1993), edge decimation (Soucy & Laurendeau, 1996) and edge collapsing (Ronfard & Rossignac, 1996) are employed to encode geometry. These mesh simplification methods progressively remove those mesh vertices which yield the smallest distortion. In order to enable the reconstruction of the original mesh at various levels of detail (LODs), the discarded vertices are encoded in the compressed bit-stream. Mesh compression systems belonging to this category include Progressive Meshes (Li & Kuo, 1998), (Pajarola & Rossignac, 2000) and Topological Surgery (Taubin et al., 1998). These techniques generally exhibit two major drawbacks: first, due to the highly irregular topology of the input mesh, a large source rate is needed for lossless encoding of connectivity. Secondly, encoding the removed vertices in the compressed bit-stream is quite costly for high-resolution meshes. Therefore, such schemes are not useful for complex meshes containing a large number of vertices. An alternative that solves the problem of the large source rates needed to encode the connectivity information, described above, is remeshing, which can be used to convert the original irregular mesh into a mesh consisting of regular elements, such as B-spline (Eck & Hoppe, 1996) or subdivision connectivity patches (Eck et al., 1995). The regular

mesh lends itself better to compression, and hence compared to the irregular mesh a much lower rate is needed to losslessly encode its connectivity information. Furthermore, multiresolution techniques alleviate the second problem of having to encode all the original vertices, because only detail information has to be encoded in order to create multiple LODs (or multiple resolution levels). Remeshing together with subdivision-based multiresolution (Lounsbery et al., 1997) are the two major components of the second category of codecs which use space-frequency dilation methods such as wavelet transforms to decorrelate the input mesh data (Khodakovsky et al., 2000), (Denis et al., 2010b). The generated wavelet coefficients are compressed using tree-based bit-plane coding methods (Shapiro, 1993), (Munteanu et al., 1999b) to achieve high compression efficiency. Multiresolution mesh compression techniques provide substantial compression gains compared to their competing schemes, and in this chapter we will confine our discussion to these techniques only.

In the recent past, several multiresolution scalable mesh compression schemes have been proposed. The majority of these schemes use coding techniques which were specifically developed for image compression. However, in general, image and mesh data exhibit different statistical characteristics as the images are consisting of pixels (with intensities) while mesh data involve geometry, i.e., the positions of vertices in a 3D space. Thus, one must be cautious when extrapolating image compression techniques towards mesh geometry encoding.

In this book chapter, we propose a constructive design methodology for multiresolution-scalable mesh compression systems. The input mesh is assumed to possess subdivision connectivity (Lounsbery et al., 1997), i.e., the connectivity in the mesh is built through subdivision[1]. A 3D mesh with subdivision connectivity is also referred to as a semi-regular mesh. With respect to *design*, we address two major aspects of scalable wavelet-based mesh compression systems, namely, the optimality of embedded quantization in scalable mesh coding and the type of coefficient dependencies that can assure the best compression performance. In this context, thorough analyses investigating the aforementioned aspects are carried out to establish the most appropriate design choices. Later on, the derived design choices are integrated as components of the scalable mesh coding system to achieve state-of-the-art compression performance.

The remainder of the book chapter is organized as follows: in Section 2, a brief overview of multiresolution analysis of the mesh geometry is given. Section 3 presents a model-based theoretical investigation of optimal embedded quantization in wavelet-based mesh coding. An information theoretic analysis of the statistical dependencies among wavelet coefficients and the conclusions regarding the best exploitable statistical dependency are detailed in Section 4. Section 5 gives an overview of the state-of-the-art mesh compression systems.

2. Multiresolution analysis of semi-regular meshes

A 3D mesh $M = \{ \mathbf{c}, \mathbf{p} \}$ is generally represented as a set of two components, a vertex list $\mathbf{c}$ and a polygon list $\mathbf{p}$. $\mathbf{c}$ is a matrix whose ith row c_i contains the x, y and z position of the ith vertex, i.e., $c_i = \left[c_{i,x}, c_{i,y}, c_{i,z} \right]$. $\mathbf{p}$ is a list of polygons made up of edges where each edge is a line connecting two vertices. In computer graphics, 3D meshes are constructed

[1] In general, an initial remeshing step (Eck et al., 1995) is required to convert the original irregular mesh into a mesh with the required connectivity.

using different polygonal shapes, e.g., triangles, rectangles etc. However, in this chapter, we will confine our discussion to the triangular meshes only.

In the following, a brief theoretical overview of semi-regular multiresolution analysis is presented. Later on, two practical transforms, namely the lifting-based wavelet transform and the spatially adapted wavelet transform are detailed.

2.1 Theory

2.1.1 Subdivision surfaces

Subdivision is a process of iteratively refining a control polyhedron M^0 into fine geometry polyhedra such that the refined polyhedra $M^1, M^2, M^3 \ldots$ converge to a limit surface M^∞. In general, subdivision schemes consist of *splitting* and *averaging* steps. In the *splitting* step, each triangular face is split into four sub-triangles by adding new vertices. This way, an intermediate polyhedron $\bar{M}^j$ is created for any level j. The *averaging* step is used to determine the position of each vertex in M^j from its local neighborhood of vertices in $\bar{M}^j$, $j = 1,2,...,J$.

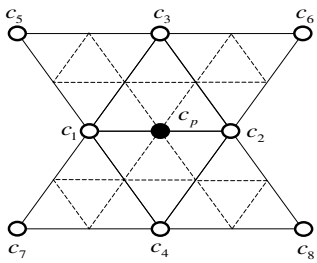


Fig. 1. Butterfly subdivision stencil.

$\mathbf{P}^j \in \mathbb{R}^{N_{j+1} \times N_j}$ and $\mathbf{Q}^j \in \mathbb{R}^{N_{j+1} \times N_{j+1}}$ (where N_j denotes the number of vertices of M^j) are the splitting and averaging matrix at level j. The subdivision process, expressed in matrix form, can be written as:

$$\mathbf{c}^{j+1} = \mathbf{Q}^j \cdot \mathbf{P}^j \cdot \mathbf{c}^j, \quad j = 0,1,2,...,J-1 .$$

A commonly used subdivision is Butterfly subdivision (Dyn et al., 1990). The subdivision stencil for Butterfly is shown in Fig. 1, where the position of a newly introduced vertex p is computed as, $c_p = \sum_{i=1}^{8} a_i c_i$ whereby a_i's denote the Butterfly weights (Dyn et al., 1990). Loop (Loop et al., 2009) and Catmull-Clark (Catmull & Clark, 1978) are among the other commonly used subdivision schemes for 3D meshes.

2.1.2 Multiresolution analysis

Lounsbery (Lounsbery et al., 1997) first invented the multiresolution analysis for arbitrary topology semi-regular surfaces using subdivision. He proved that refinable bases exist when a coarse mesh M^0 is refined through subdivision, i.e.,

$$\boldsymbol{\phi}^j(\mathbf{x}) = \boldsymbol{\phi}^{j+1}(\mathbf{x}) \cdot \mathbf{P}^j, \text{ for } \mathbf{x} \in M^0 \text{ and } 0 \le j < J. \tag{1}$$

$\boldsymbol{\phi}^j(\mathbf{x})$ in the above equation denotes the row vector of scaling functions ϕ_i^j. Given these refinable scaling functions, scalar-valued function spaces associated with the coarsest geometry M^0 are defined as (Lounsbery et al., 1997):

$$V^j\left(M^0\right) := \text{Span}\left(\boldsymbol{\phi}^j(\mathbf{x})\right), \text{ for } 0 \le j < J. \tag{2}$$

Eq (1) implies that these spaces are indeed nested, i.e.,

$$V^0\left(M^0\right) \subset V^1\left(M^0\right) \subset V^2\left(M^0\right) \subset \ldots, \tag{3}$$

The wavelet space $W^j\left(M^0\right)$ is defined as a space which is the orthogonal complement of $V^j\left(M^0\right)$ in $V^{j+1}\left(M^0\right)$. Hence, $W^j\left(M^0\right)$ and $V^j\left(M^0\right)$ together can represent any scalar-valued piecewise function in the space $V^{j+1}\left(M^0\right)$. If $\boldsymbol{\psi}^j(\mathrm{x})$ is a row vector containing refinable bases functions of $W^j\left(M^0\right)$, the following stands (Lounsbery et al., 1997):

$$\boldsymbol{\psi}^j(\mathbf{x}) = \boldsymbol{\phi}^{j+1}(\mathbf{x}) \cdot \mathbf{Q}^j, \text{ for } \mathbf{x} \in M^0 \text{ and } 0 \le j < J. \tag{4}$$

Combining (1) with (4) yields

$$\left(\boldsymbol{\phi}^j(\mathbf{x}), \boldsymbol{\psi}^j(\mathbf{x})\right) = \boldsymbol{\phi}^{j+1}(\mathbf{x}) \cdot \left(\mathbf{P}^j, \mathbf{Q}^j\right), \text{ or } \left(\boldsymbol{\phi}^j(\mathbf{x}), \boldsymbol{\psi}^j(\mathbf{x})\right) \cdot \left(\mathbf{P}^j, \mathbf{Q}^j\right)^{-1} = \boldsymbol{\phi}^{j+1}(\mathbf{x}). \tag{5}$$

A set of scaling functions $\boldsymbol{\phi}^{j+1}(\mathbf{x})$ can then be used to decompose a surface S^{j+1} in $V^{j+1}(M^0)$, i.e.,

$$S^{j+1} = \sum_i c_i^{j+1} \phi_i^{j+1} = \boldsymbol{\phi}^{j+1}(\mathbf{x}) \cdot \mathbf{c}^{j+1}, \tag{6}$$

where c_i^{j+1} is the ith vertex in M^{j+1}. Since the analysis filters are uniquely determined by the relationship

$$\left(\mathbf{P}^j \middle| \mathbf{Q}^j\right)^{-1} = \begin{pmatrix} \mathbf{A}^j \\ \mathbf{B}^j \end{pmatrix}, \tag{7}$$

by combining Eq (5) and Eq (6) and making the above substitution for $\left(\mathbf{P}^j \middle| \mathbf{Q}^j\right)^{-1}$, we obtain:

$$S^{j+1} = \boldsymbol{\phi}^j(\mathbf{x}) \cdot \mathbf{A}^j \cdot \mathbf{c}^{j+1} + \boldsymbol{\psi}^j(\mathbf{x}) \cdot \mathbf{B}^j \cdot \mathbf{c}^{j+1}. \tag{8}$$

From Eq (8), Eq (9) one derives the forward wavelet transform, given by:

$$\mathbf{c}^j = \mathbf{A}^j \cdot \mathbf{c}^{j+1}, \quad \mathbf{d}^j = \mathbf{B}^j \cdot \mathbf{c}^{j+1}, \ \forall j: \ 0 \le j < J, \tag{9}$$

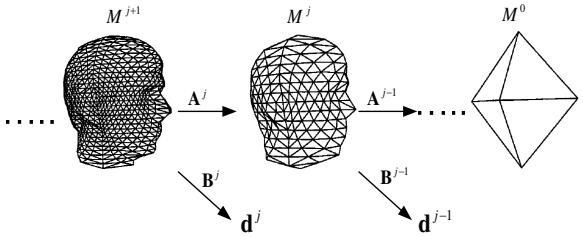


Fig. 2. Pictorial representation of the forward wavelet decomposition, (Lounsbery et al., 1997).

where $\mathbf{d}^j$ is a matrix containing the wavelet coefficients for the jth level of the transform. In general, after the transform, a fair amount of correlation still exist between x, y and z wavelet component. Local frame representation (Khodakovsky et al., 2000) of wavelet coefficients is often used to make wavelet components much more independent. After the local frame transformation, each wavelet coefficient consists of a *normal* and two *tangential* components. $\mathbf{A}^j$s and $\mathbf{B}^j$s are matrices representing the low and the high-pass filters, respectively, also referred to as analysis filter pairs.

A similar reasoning as for Eq (9) can be used to formulate the inverse wavelet transform, expressed by:

$$\mathbf{c}^{j+1} = \mathbf{P}^j \cdot \mathbf{c}^j + \mathbf{Q}^j \cdot \mathbf{d}^j, \text{ for } \forall j:\ 0 \le j < J. \tag{10}$$

Hence, $\mathbf{P}^j$s and $\mathbf{Q}^j$s jointly form the synthesis part of the decomposition for the lossless reconstruction of the input semi-regular mesh M^J. Note that the computation of the $\mathbf{A}^j$s and $\mathbf{B}^j$s involves the inversion of a large matrix, which makes the forward transform more complex than the inverse transform.

2.2 Lifting-based wavelet transform

As explained earlier, the filter bank implementation of multiresolution analysis is quite complex in the sense that the computation of analysis filters involve the computationally intensive inversion of large subdivision matrices. In this context, the lifting-based wavelet implementation (Schröder & Sweldens, 1995) provides a low complexity construction of multiresolution methods. In lifting-based multiresolution analysis, each scaling function ϕ_i^j of the jth level exists so that $\{\phi_i^j \mid i \in M^j\}$ is a Riesz basis of $V^j\left(M^0\right)$ (Schröder & Sweldens, 1995). The refinement relation for the scaling functions is then:

$$\phi_i^{\,j} = \sum_l p_{i,l}^j \cdot \phi_i^{\,j+1}, \tag{11}$$

where l is the set which defines all linear combination of scaling functions and $p_{i,l}^j$ forms the entries of a matrix similar to $\mathbf{P}^j$. A similar refinement relation as Eq (11) is also defined for wavelet functions, i.e., each wavelet function $\psi_k^{\,j}$ exists so that $\{\psi_k^{\,j} \mid k \in K^j\}$ is a Riesz basis of $W^j\left(M^0\right)$:

$$\psi_k{}^j = \sum_l q_{k,l}^j \cdot \phi_k{}^{j+1}. \tag{12}$$

K^j and M^j are disjoint sets and they jointly form the scaling function index set of the next higher level, i.e., $M^{j+1} = M^j \oplus K^j$. The lifting-based forward decomposition is expressed by the following relations (Schröder & Sweldens, 1995):

$$\begin{aligned}
&\forall i \in M^j: \quad c_i{}^j = c_i{}^{j+1} \Rightarrow \text{subsample} \\
&\forall k \in K^j: \quad d_k{}^j = c_k{}^{j+1} - \sum_{i \in M^j} a_i \, . c_i{}^j \Rightarrow \text{prediction} \\
&\forall k \in K^j: \begin{cases} c_1^j = c_1^j + \tilde{a}_1 \cdot d_k{}^j \\ c_2^j = c_2^j + \tilde{a}_2 \cdot d_k{}^j \end{cases} \Rightarrow \text{update}
\end{aligned} \tag{13}$$

In the forward transform, the first step is to produce a lower-resolution mesh M^j starting from a higher-resolution version M^{j+1}. The wavelet coefficient $d_k{}^j$ is the *prediction error* when a high-resolution vertex $c_k{}^{j+1}$ is predicted based on its low-resolution neighborhood in M^j. After the prediction, an *update* step is used to modify the low resolution mesh M^j. The *update* step is carried out on a pair $\{c_1, c_2\}$ of low-resolution vertices joined by a parent edge (Schröder & Sweldens, 1995) using the update weights $\{\tilde{a}_1, \tilde{a}_2\}$. In general, the prediction and update weights only depend on the connectivity with respect to the vertex to be predicted. However, specific multiresolution analyses for which the weights depend on the specific resolution level and the underlying geometry can be also constructed (more details are given in Section 2.3).

The inverse transform can be formulated by following the forward-transform steps in the reverse order, i.e.:

$$\begin{aligned}
&\forall k \in K^j: \begin{cases} c_1^j = c_1^j - \tilde{a}_1 \cdot d_k{}^j \\ c_2^j = c_2^j - \tilde{a}_2 \cdot d_k{}^j \end{cases} \Rightarrow \text{inverse update} \\
&\forall k \in K^j: \quad c_k{}^{j+1} = d_k{}^j + \sum_{i \in M^j} a_i \cdot c_i{}^j \Rightarrow \text{inverse predict} \\
&\forall i \in M^j: \quad c_i{}^{j+1} = c_i{}^j \Rightarrow \text{inverse subsample}
\end{aligned} \tag{14}$$

2.3 Spatially Adaptive Wavelet Transform (SAWT)

As mentioned earlier, lifting-based transforms generally employ fixed prediction weights, independent of the spatial position and geometry around the vertex to be predicted. A simple observation reveals that a better prediction can result from adapting the prediction to the underlying geometry of the mesh. This argument is explained with a simple example: Fig. 3, referring to the position variable of the vertices, shows a scenario where the vertex to be predicted c_p lies on the straight line joining the vertex pair $\{c_1, c_2\}$, while the remaining six coarser vertices $\{c_i\}_{i=3}^8$ lie on two different planes. In this situation, a prediction function for c_p involving all eight coarser vertices will not be optimal and a better prediction could result by using c_1 and c_2 only. This is logical since c_p lies on the edge formed by the vertex pair $\{c_1, c_2\}$ and is geometrically more correlated to vertices $\{c_1, c_2\}$. Thus, an efficient

prediction can be achieved if the prediction process is adapted to the local mesh geometry. Efficient prediction results in smaller energy of wavelet coefficients and hence an improved compression performance of the mesh coding system. To reverse the prediction operation, the decoder needs to know the weights used by the encoder for the prediction of each vertex c_p. Since additional rate (compared to classical Butterfly) needs to be spent for coding the prediction weights, the total compression efficiency in the geometry adaptive case is a compromise between the bitrate saved due to the efficient prediction and the extra bitrate needed for signaling the prediction weights.

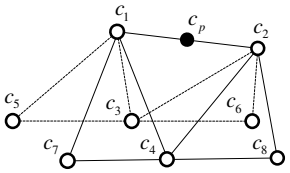


Fig. 3. Butterfly footprint on an edge.

In the following, a finite set of prediction filters is proposed in the context of spatially-adaptive wavelet transforms (SAWT) (Denis et al., 2010a). The idea is to use one filter out of this set which best suits the geometry around the vertex to be predicted and which results in the smallest prediction error. A careful application of such an adaptive approach will provide an average rate gain if the reduction in the bitrate due to better prediction dominates the extra bitrate needed to signal the filter type to the decoder.

In a first step, the input semi-regular mesh is segmented into regions as follows. Let $B(r,s)$ denote the bounding box of the input semi-regular mesh, where $r=(x_B,y_B,z_B)$ and $s=(s_x,s_y,s_z)$ represent the coordinates of the top-left corner and the size vector, respectively. Considering the bounding box as the root cell, each cell on a certain tree level is recursively split into eight equally sized sub-cells to create the next level of the octree. This recursive splitting continues until the number of vertices in the highest-level cells are smaller than a user-defined threshold α. This way, the semi-regular mesh is divided into regions of approximately the same size – see Fig. 4.

For each region k, the wavelet analysis is performed by selecting one of the six candidates filters given below:

$$f_1=\tfrac{1}{2}(c_1+c_2)+\tfrac{1}{8}(c_3+c_4)-\tfrac{1}{16}(c_5+c_6+c_7+c_8)\Rightarrow(\text{Butterfly})$$
$$f_2=\tfrac{1}{2}(c_1+c_2)+\tfrac{1}{4}(c_3+c_4)-\tfrac{1}{8}(c_5+c_6+c_7+c_8)\Rightarrow(\text{Modified Butterfly})$$
$$f_3=\tfrac{3}{8}(c_1+c_2)+\tfrac{1}{8}(c_3+c_4)\Rightarrow(\text{Loop})$$
$$f_4=\tfrac{1}{2}(c_1+c_2)\Rightarrow(\text{edge})$$
$$f_5=\tfrac{1}{2}(c_3+c_4)\Rightarrow(\text{anti edge})$$
$$f_6=\tfrac{1}{4}(c_1+c_2+c_3+c_4)\Rightarrow(\text{Hybrid of } f_4 \text{ and } f_5)$$

Note that the above set of filters is defined using a mixture of Butterfly, Loop and midpoint subdivision schemes.

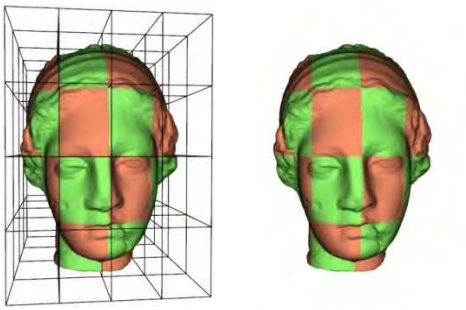


Fig. 4. Mesh partitioning for $\alpha = 400$. The red and green patches indicate different regions for which different prediction filters will be selected.

Similar to (Chang & Girod, 2006), a filter candidate for a particular region k in M^l , is chosen in an optimal distortion-rate (D-R) manner. More specifically, a predictor for each region k is selected such that the following Lagrangian cost function is minimized:

$$\Lambda_{M^l,k} = \underset{l\in\{1,2,..,6\}}{\arg\min}\left\{E_k\left[\left(c_p - \tilde{c}_{p,f_l}\right)^2\right] + \lambda \cdot R_k\left(f_l\right)\right\} \tag{15}$$

where $R_k\left(f_l\right)$ denotes the rate necessary for encoding the filter index l used for prediction in the region *k*.

3. Scalable quantization of wavelet coefficients

In scalable mesh compression, the wavelet coefficients in the subbands are quantized using a generic family of embedded deadzone scalar quantizers (EDSQ) (Taubman & Marcelin, 2001), in which every wavelet coefficient X is quantized to:

$$q_{\xi,n} = \begin{cases} sign(X)\cdot\left\lfloor \dfrac{|X|}{\Delta_n} + \xi_n \right\rfloor & if\ \dfrac{|X|}{\Delta_n} + \xi_n > 0 \\ 0 & otherwise \end{cases} \tag{16}$$

where $n \in \mathbb{Z}_+$ denotes the quantization level. ξ_n and Δ_n denote the deadzone control parameter and the step size for any $n \geq 0$, respectively, with $\xi_n = \xi_0 / 2^n$ and $\Delta_n = 2^n \Delta_0$, where ξ_0 and Δ_0 are the parameters for the highest rate quantizer $(n = 0)$. Note that $\xi_0 = 0$ corresponds to the well-known SAQ (Shapiro, 1993) in which the deadzone size is twice the step size Δ_n for any n .

3.1 Wavelet coefficient histogram

In general, the observed histogram H_k^j of the kth , $k \in \{x, y, z\}$, coordinate component of the jth wavelet subband is symmetric around its center of mass which is often zero or very close to zero. Moreover, the histogram is peaky around the mean and the frequency of occurrence decays as the magnitude of the coefficient's component increases. Fig. 5 depicts

the observed histograms of the d^{J-3} subband of *Rabbit* (*non-normal* mesh) and *Dino* (*normal* mesh) obtained using the classical Butterly transform. It is observed experimentally that, in general, $\sigma^2(H_k^{j+1}) < \sigma^2(H_k^j)$ for $1 \le j < J$.

In the literature, the observed histogram of any component of a wavelet subband is generally modeled using a zero mean generalized Gaussian (GG) distribution (Mallat, 1989), expressed by:

$$\forall x \in \mathbb{R} \quad f_{GG}(x,\sigma,\alpha) = \frac{\alpha \upsilon^{1/\alpha}}{2\Gamma(1/\alpha)} e^{-\upsilon|x|^{\alpha}}, \tag{17}$$

where α, $\alpha \in (0,2]$, is the shape control parameter. $\upsilon > 0$ is the scaling factor and $\upsilon^{1/\alpha} = \sqrt{\Gamma(3/\alpha)/\sigma^2\Gamma(1/\alpha)}$, where Γ is the Gamma function. Note that, for $\alpha = 1$, Eq (17) transforms into a zero-mean Laplacian probability density function (PDF) given by:

$$\forall x \in \mathbb{R} \quad f_L(x,\sigma) = \frac{1}{\sigma\sqrt{2}} e^{-\frac{\sqrt{2}}{\sigma}|x|} = \frac{\lambda}{2} e^{-\lambda|x|} \text{ where } \lambda = \frac{\sqrt{2}}{\sigma}, \tag{18}$$

and for $\alpha = 2$ Eq (17) corresponds to a zero-mean Gaussian PDF.

Although GG distributions closely approximate the observed histogram of wavelet coefficients, only approximate rate and distortion expressions for a uniformly quantized GG random variable are known (Fraysse et al,. 2008). The extension of these expressions to embedded quantization is not evident as the rate and distortion functions for such distributions are not easily tractable and can only be computed numerically. Moreover, computing these quantities gets very cumbersome due to the slow numerical integration of expressions involving a GG probability function, especially for $\alpha << 1$.

3.2 Proposed Laplacian mixture model

In order to avoid the aforementioned drawbacks of GG distributions, we propose a simple Laplacian mixture (LM) model which not only gives an easy closed-form derivation of the distortion and rate quantities but also better approximates the observed histogram of wavelet coefficients in the majority of cases. The proposed LM is a linear combination of two Laplacian PDFs, i.e.,

$$\forall x \in \mathbb{R} \quad f_{LM}(x) = \beta \cdot f_L(x,\sigma_1) + (1-\beta) \cdot f_L(x,\sigma_2). \tag{19}$$

Note that $f_{LM}(x)$ indeed defines a probability function, as $\int_{-\infty}^{\infty} f_{LM}(x)\,dx = 1$.

The LM model is fitted over the observed data using the expectation maximization (EM) algorithm (Dempster et al., 1977) in order to determine the parameters σ_1, σ_2 and β. The E-step in the EM process calculates two responsibility factors

$$r_1(i) = \frac{\beta \cdot f_L(x_i,\sigma_1)}{\beta \cdot f_L(x_i,\sigma_1) + (1-\beta) \cdot f_L(x_i,\sigma_2)}, \quad r_2(i) = \frac{(1-\beta) \cdot f_L(x_i,\sigma_2)}{\beta \cdot f_L(x_i,\sigma_1) + (1-\beta) \cdot f_L(x_i,\sigma_2)},$$

of each observation $x_i, 1 \le i \le N$ and the M-step updates the parameters to be estimated, as:

$$\sigma_m = \sqrt{2}\frac{\sum_{i=1}^{N} r_m(i).|x_i|}{\sum_{i=1}^{N} r_m(i)}, \quad m = 1,2, \quad \text{and} \quad \beta = \frac{1}{N}\sum_{i=1}^{N} r_1(i).$$

The E- and M- steps are executed in tandem till the algorithm achieves minimum *Kullback-Leibler* (KL) distance between the observed and model histograms. A better convergence rate is achieved by the initialization condition $\sigma_1^2 = 0.5\sigma_E^2$, $\sigma_2^2 = 2\sigma_E^2$ and $\beta = 0.9$, where σ_E^2 is the estimated data variance. Histogram fitting for GG distributions is done using the brute-force method where parameters σ_1, σ_2 and β are exhaustively computed for a minimum KL distance.

3.3 Distortion-Rate (D-R) function

Closed-form expressions for the output distortion D_L and the output rate R_L of a Laplacian source quantized using an n level EDSQ are derived in the Appendix. In this section, we derive the D-R function for our proposed LM model. Since the distortion is a linear function of the source PDF, the output distortion D_{LM} of the LM PDF for any quantization level n can be written as:

$$D_{LM}\left(Q_{\delta_n,\Delta_n}\right) = \beta \cdot D_L\left(Q_{\delta_n,\Delta_n}\right) + (1-\beta)\cdot D_L\left(Q_{\delta_n,\Delta_n}\right), \text{ with } \delta_n = 1 - \xi_n. \tag{20}$$

This does not hold for the output rate R_{LM} since the entropy involves the non-linear $\log(.)$ function. Instead, R_{LM} can be computed as an infinite sum:

$$P_0 = 2\int_0^{\delta_n\Delta_n} f_{LM}(x)dx,\ P_k = \int_{(k-1+\delta_n)\Delta_n}^{(k+\delta_n)\Delta_n} f_{LM}(x)dx,\ k = 1,2,3..., \text{ and } R_{LM}(Q_{\delta_n,\Delta_n}) = -\sum_{k=-\infty}^{\infty} P_k \log_2 P_k$$

,

where P_k denotes the probability mass of the kth quantization cell ($k = 0$ corresponds to the deadzone cell). Since the LM model is symmetric around its mean, $P_k = P_{-k}$. Note that the probability mass function (PMF) can be computed exactly due to the possibility of analytical integration of $f_{LM}(x)$. For the GG distribution, however, only numerical integration is possible.

3.4 Model validation

This section demonstrates that the proposed LM model is able to approximate the observed histogram and the observed D-R function of 3D wavelet coefficients more accurately compared to the commonly utilized GG distributions. For comparison purpose, results for the single Laplacian $f_{LM}(\beta = 0)$ case are also reported.

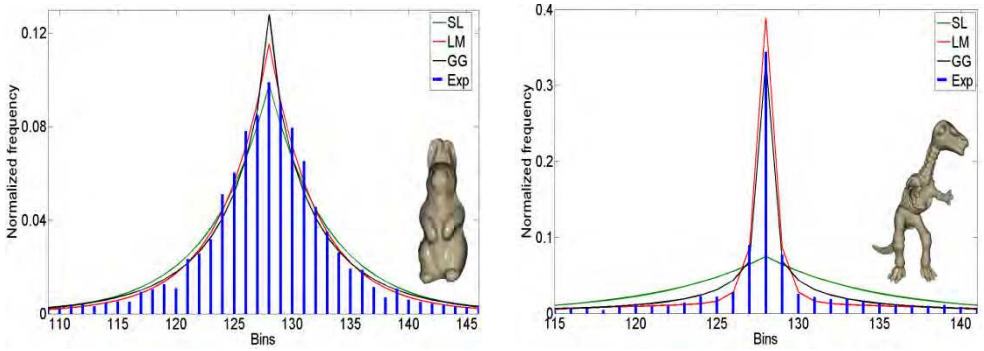


Fig. 5. Probability function fitting over the observed histogram (Exp) for $\mathbf{d}^{J-3}$ *-normal component* for Rabbit (left) and Dino (right). SL is used as the abbreviation of single Laplacian PDF.

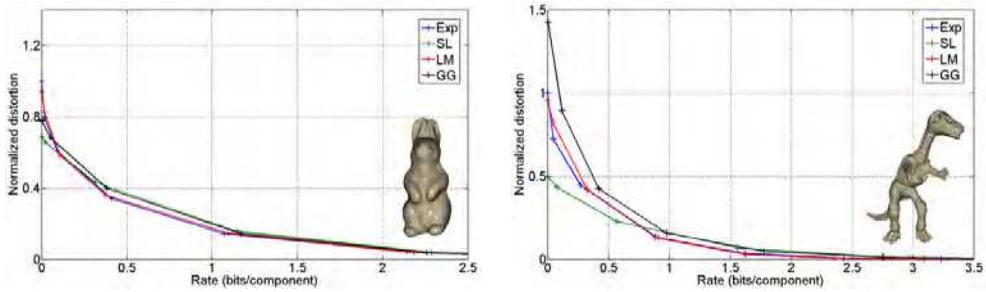


Fig. 6. Modeled and observed D-R functions for the histograms of Fig. 5. Rate is taken as bits per spatial coordinate component.

Fig. 5 illustrates that the proposed mixture model provides a better fitting probability function for the observed histogram compared to the Laplacian and GG distributions. This is especially true for the middle range positive and negative coefficients values – see Fig. 5. For the Rabbit mesh, LM gives only slightly better fitting than the other two models. However, for Dino, the LM can clearly model the fast decay of the observed histogram more accurately than the GG. The Laplacian PDF in this case only gives a very coarse approximation of the observed histogram.

Mesh Type	Mesh (Filter)	SL			LM			GG		
		Nor	*Tan 1*	*Tan 2*	*Nor*	*Tan 1*	*Tan 2*	*Nor*	*Tan1*	*Tan 2*
Non-Normal	*Venus(U-BF)*	0.097 (6.3)	0.114 (10.3)	0.103 (8.9)	0.075 (1.3)	0.100 (3.3)	0.091 (2.8)	0.086 (4.7)	0.102 (5.0)	0.089 (4.7)
	Venus(L-BF)	0.137 (11.4)	0.137 (10.1)	0.104 (6.0)	0.090 (2.0)	0.108 (2.9)	0.080 (1.9)	0.112 (8.1)	0.121 (5.4)	0.092 (4.1)
	Venus(Loop)	0.113 (9.4)	0.102 (7.3)	0.091 (6.7)	0.085 (3.3)	0.090 (1.8)	0.069 (1.7)	0.098 (7.1)	0.092 (3.9)	0.081 (5.0)
	Rabbit(U-BF)	0.170 (8.6)	0.171 (10.4)	0.172 (10.2)	0.134 (1.4)	0.136 (2.0)	0.132 (1.8)	0.150 (5.7)	0.143 (5.2)	0.147 (6.2)
	Rabbit(L-BF)	0.208 (12.0)	0.188 (10.7)	0.177 (8.4)	0.143 (2.5)	0.140 (1.5)	0.138 (1.8)	0.160 (6.7)	0.152 (5.1)	0.153 (5.3)
	Rabbit(Loop)	0.167 (11.2)	0.207 (11.2)	0.173 (8.3)	0.115 (2.4)	0.156 (1.8)	0.135 (2.3)	0.136 (7.9)	0.177 (7.6)	0.152 (5.2)
Normal	*Dino(U-BF)*	0.527 (16.2)	0.656 (34.3)	0.971 (42.7)	0.145 (5.6)	0.147 (7.8)	0.154 (9.4)	0.165 (7.5)	0.132 (23.4)	0.141 (30.3)
	Skull(U-BF)	1.108 (37.2)	1.473 (44.9)	1.877 (50.4)	0.120 (3.9)	0.138 (7.9)	0.157 (20.5)	0.145 (12.8)	0.141 (15.4)	0.141 (19.9)
	Skrewdriver(U-BF)	0.5294 (33.0)	0.6477 (42.4)	0.6377 (41.9)	0.309 (14.4)	0.251 (17.0)	0.263 (20.0)	0.315 (25.0)	0.262 (34.8)	0.249 (35.2)

Table 1. *KL* (%*ME*, the modeling error as defined in Eq (19)) for the *normal* (*NOR*) and the two *tangential* components (*TAN1, TAN2*) averaged over the five subbands. *U-BF* (Unlifted Butterfly), *L-BF* (Lifted Butterfly).

Fig. 6 plots the observed and model D-R curves for the same subband as the one used in Fig. 5. For Rabbit, the LM D-R almost completely overlaps the observed D-R curve. In both cases, the D-R function of the proposed LM model follows the experimental D-R curve more closely than the other two models.

In Table 1, the average *KL* divergence results for the Laplacian, GG and LM models for two non-normal (Venus, Rabbit) and three normal (Dino, Skull, Skredriver) meshes are shown. Each of the three coordinate components is considered separately. For each trial of Table 1, average is taken over five highest resolution subbands. For the large majority of cases, the LM model gives better fitting of the observed histogram than the competing GG model. Note that the Laplacian model gives always the worst fitting results. Also, the LM model gives equally good fitting for both *normal* (*Nor*) and *tangential* (*Tan 1* and *Tan 2*) components. Superior histogram fitting results of our proposed model are also observed for the SAWT of Section 2.3. These results are not reported here due to lack of space.

In Table 1, the percentage modeling error $ME(\%)$ relative to the *KL* divergence is shown in parenthesis of each table entry. The $ME(\%)$ is defined in order to gauge the D-R accuracy of the proposed mixture model with respect to other two models. $ME(\%)$ is defined as:

$$ME(\%) = \frac{\int_{R\in\Re} \left| D_M(R) - D_E(R) \right|}{\int_{R\in\Re} \max_R \left\{ D_M(R), D_E(R) \right\}} \times 100 \,. \tag{21}$$

Mesh Type	Mesh (Filter)	SL			LM			GG		
		$J-1$	$J-2$	$J-3$	$J-1$	$J-2$	$J-3$	$J-1$	$J-2$	$J-3$
Non-Normal	*Venus(U-BF)*	0.044 (13.6)	0.038 (8.4)	0.039 (3.5)	0.014 (2.9)	0.008 (0.85)	0.025 (1.5)	0.027 (9.5)	0.024 (5.6)	0.033 (2.8)
	Venus(L-BF)	0.050 (13.8)	0.070 (13.9)	0.076 (7.6)	0.010 (2.1)	0.010 (1.5)	0.027 (1.7)	0.029 (9.9)	0.041 (10.1)	0.049 (4.1)
	Venus(Loop)	0.051 (13.8)	0.054 (11.0)	0.038 (4.0)	0.016 (2.4)	0.009 (1.6)	0.023 (0.80)	0.036 (11.0)	0.031 (6.8)	0.032 (2.7)
	Rabbit(U-BF)	0.064 (14.0)	0.062 (11.4)	0.082 (8.1)	0.008 (1.5)	0.011 (1.0)	0.035 (1.3)	0.029 (8.6)	0.032 (6.8)	0.054 (4.9)
	Rabbit(L-BF)	0.069 (14.2)	0.093 (13.9)	0.111 (11.2)	0.007 (1.5)	0.011 (1.1)	0.035 (1.8)	0.029 (8.6)	0.040 (8.6)	0.058 (5.9)
	Rabbit(Loop)	0.082 (16.4)	0.088 (14.7)	0.085 (9.0)	0.011 (2.0)	0.013 (1.9)	0.034 (2.0)	0.042 (11.5)	0.038 (8.7)	0.058 (5.4)
Normal	*Dino(U-BF)*	1.208 (56.7)	0.873 (46.7)	0.623 (34.7)	0.029 (13.7)	0.074 (12.8)	0.049 (4.6)	0.031 (41.3)	0.042 (32.9)	0.058 (20.4)
	Skull(U-BF)	2.039 (65.7)	1.981 (49.9)	1.832 (32.4)	0.054 (34.2)	0.040 (7.6)	0.076 (5.1)	0.036 (40.2)	0.068 (15.1)	0.066 (8.3)
	Skrewdriver(U-BF)	0.536 (67.0)	0.696 (61.6)	0.483 (39.7)	0.067 (53.2)	0.074 (18.2)	0.065 (7.3)	0.035 (64.0)	0.064 (54.3)	0.101 (25.4)

Table 2. *KL* (%*ME*) for three resolution subbands averaged over the three coordinate components.

From Table 1, it is evident that on average the proposed LM model performs better than the GG and Laplacian models also in the *ME* sense. Better *ME* results are also obtained for SAWT (not reported here). Hence, the proposed LM model along with the derived D-R function is a better choice for modeling both the histogram and the D-R curve of mesh wavelet coefficients compared to the contemporary models. One notices that, a best histogram fitting in *KL* sense may not always yield the lowest *ME*.

Table 2 reports the model validation results for different resolution subbands. For each trial the average is taken across the three spatial coordinate components. It is observed that the GG model performs slightly better for the low-resolution subbands of some meshes. The observed histograms in such cases are more Gaussian-alike, i.e., they have a round top. In general, the LM model faces difficulty in approximating such a round-top histogram due to the peaky nature of each of its Laplacian components; the GG fits well such histograms, as it corresponds to a Gaussian distribution for $\alpha = 2$. Nevertheless, the results show that, on average, the LM model outperforms the Laplacian and the GG models in *KL* as well as in *ME* sense.

3.5 Optimal embedded quantization

In this section, conclusions regarding the optimal EDSQ to be used in scalable wavelet-based coding of meshes are drawn. Let z denote the ratio between the deadzone size for $n = 0$ (see Eq. (16)) and the step size for $n \geq 0$ of a general EDSQ. The total average signal-to-noise

ratio (*SNR*) difference which is utilized to measure the performance gap of different embedded quantizers is defined as:

$$\Delta\overline{SNR} = \frac{1}{N}\sum_{\Re}\left(\underset{z=1}{SNR(R)} - \underset{z>1}{SNR(R)} \right),$$

which is computed over a rate range $\Re$ for N rate points, where $SNR(R)$ denotes the discrete *SNR*-rate function. The $SNR = 10\log_{10}(\sigma^2/D)$ is computed in dBs, where D is the total distortion in the transform domain. The difference in *SNR* is computed relative to the uniform embedded quantizer (UEQ), i.e., $z=1$. $\Delta\overline{SNR}$ for five embedded deadzone quantizers is plotted in Fig. 7. over a wide range of standard deviation ratios σ_2/σ_1. In Fig. 7., the commonly observed proportion $\beta = 0.9$ is considered, as mentioned in Section 3.2.

We determined experimentally that at lower standard deviation ratios, $\Delta\overline{SNR}$ is positive and the UEQ is optimal for $\sigma_2/\sigma_1 < 120$. For $120 < \sigma_2/\sigma_1 < 290$, the quantizer with $z = 1.5$ performs better compared to all other quantizers. Similarly, $z=2$ (i.e. the SAQ) performs the best in the range $290 < \sigma_2/\sigma_1 < 600$, while $z = 2.5$ performs the best for $600 < \sigma_2/\sigma_1$. In general, small standard deviation ratios correspond to α close to 1, observed in non-normal meshes, while higher ratios correspond to $\alpha \ll 1$, observed in normal meshes. These results show that one cannot determine a single embedded quantizer that provides the best performance for *all* 3D meshes. However, an optimal quantizer per wavelet coordinate can be determined based on the corresponding σ_2/σ_1 extracted from the model.

Overall, for $\sigma_2/\sigma_1 < 120$, the difference between SAQ and the UEQ is significant, and hence UEQ is the optimal choice. For $\sigma_2/\sigma_1 \geq 120$, SAQ is not always the optimum, but lies not far from the optimum.

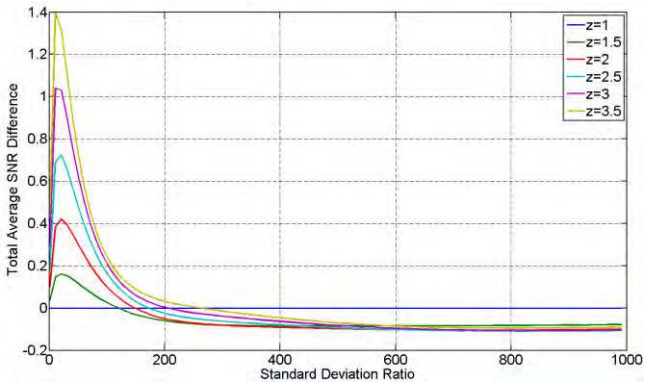


Fig. 7. *SNR* difference for five EDSQs with respect to UEQ.

Given the fact that SAQ is closely linked to bit-plane coding and that it can be implemented using simple binary arithmetic, one concludes that SAQ is not an optimal, but an acceptable solution in scalable coding of meshes.

4. Analysis of wavelet coefficient dependencies

Similar to images, parent-children and neighboring wavelet coefficient dependencies exist in wavelet decomposed mesh structure. In Fig. 8 (middle, right), the positions of the wavelet coefficients at different levels of the transform are shown with the help of white and dark circles. In particular, wavelet coefficients have a one-to-one correspondence with the edges of the coarser mesh. For each wavelet coefficient there are rings of neighboring coefficients which lie in the same wavelet subband – see Fig. 8 (right). Also, a set of four wavelet coefficients have a parent coefficient at the next coarser resolution – see Fig. 8 (middle, right).

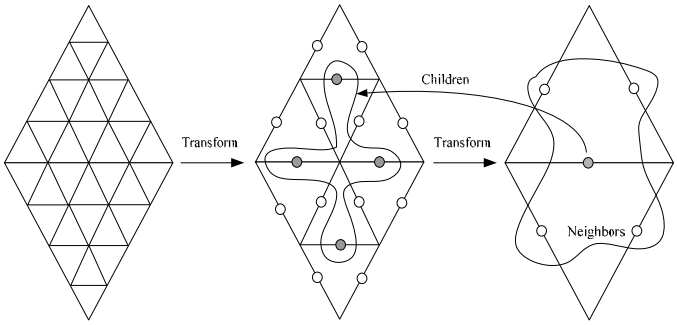


Fig. 8. Parent-children and neighboring wavelet coefficients: actual mesh (left); coarser meshes after one (middle), and after two wavelet decomposition levels (right).

Statistical intraband dependencies exist between neighboring coefficients of each resolution level. The main reason for the existence of these dependencies is the smoothness of the surface. Wavelet coding paradigms that exploit the intraband dependencies between the wavelet coefficients are known as intraband wavelet codecs such as block-based coding techniques (Munteanu et al., 1999a), quadtree coding approaches (Munteanu et al., 1999b), and the EBCOT codec employed in the JPEG-2000 scalable image coding standard (Taubman, 2000).

Statistical dependencies also exist between the parent and descendants (children) due to the natural decay of the coefficients' magnitude for increasing frequencies. In other words, if a parent coefficient magnitude is below a certain threshold, then there is a high probability that the magnitude of its descendants will be also below this threshold. This corresponds to the so-called zerotree-model, firstly introduced by Shapiro in (Shapiro, 1993). The wavelet coding paradigms that exploit the parent-children dependencies are known as interband wavelet codecs.

Finally, there is a third category of coding paradigms, exploiting both the interband and intraband statistical dependencies between the wavelet coefficients. They are generally known as composite codecs, EZBC (Hsiang & Woods, 2000) and the ECECOW approach of (Wu, 1997) are typical examples of codecs in this category.

In the following, an information theoretical analysis of the aforementioned coefficient dependencies is presented. Our aim is to single out the type of dependencies which can ensure best compression performance in the context of wavelet-based mesh compression.

4.1 Mutual information analysis

The mutual information is the reduction in the entropy of one random variable due to the knowledge of the other random variable.

$$I(X;Y) = h(X) - h(X/Y) \ , \tag{22}$$

It is known that $I(X,Y) = I(Y,X)$. In the wavelet domain, we define the following mutual information quantities:

$I(X;P_X)$: denotes the mutual information between a wavelet coefficient X and its parent coefficient P_X.

$I(X;\mathbf{n}_X)$: denotes the mutual information between a wavelet coefficient X and its neighboring wavelet coefficients $\mathbf{n}_X = [n_{1,X}, n_{2,X}, ...n_{N,X}]$.

$I(X;P_X;\mathbf{n}_X)$: denotes the composite mutual information.

From the basics of information theory (Cover & Thomas, 1991), we know that:

$$I(X;P_X;\mathbf{n}_X) \geq I(X;\mathbf{n}_X) \quad \text{and} \quad I(X;P_X;\mathbf{n}_X) \geq I(X;P_X). \tag{23}$$

For the estimation of $I(X;\mathbf{n}_X)$, we need to estimate the joint PDF $p(x,\boldsymbol{n}_x)$ which can have high dimensionality depending on the number of considered neighbors. Since the amount of data needed to accurately estimate a PDF increases exponentially with its dimensionality, it is difficult to reliably estimate a high-dimensional PDF. To alleviate this problem, the reduction in dimensionality as proposed in (Liu & Moulin, 2000) is used here. We summarize the neighborhood of X through a so-called summarizing function $T = g(\mathbf{n}_X)$. This function maps the neighboring wavelet coefficients to a single value. We note that such a many-to-one summarizing function cannot increase the mutual information, i.e.,

$$I(X;\mathbf{n}_X) \geq I(X;T) \ . \tag{24}$$

Equality in the above equation holds if $\mathbf{n}_X \to T \to X$ forms a Markov chain. The summarizing function used in our analysis is:

$$T = f(\mathbf{n}_X) = \sum_{i=1}^{N} \left(n_{i,X}\right)^2. \tag{25}$$

Due to this summarizing function, it is sufficient to compute the joint PDF $p(x,t)$, t is a realization of the random variable T, instead of $p(x,\boldsymbol{n}_x)$, for the estimation of the intraband mutual information $I(X;\mathbf{n}_X)$.

In our analysis, the mutual information for the defined quantities is estimated using the adaptive partitioning method (Darbellay & Vajda, 1999) instead of the traditional histogram method. This is because the histogram method highly depends on the bin size and for a small bin size there may not be sufficient number of observations in some bins to make a correct estimate. The adaptive partitioning method (Darbellay & Vajda, 1999) on the other

hand, ensures that there are always sufficient numbers of observations in each bin, and provides reliable estimates of the mutual information.

		Butterfly			Loop		
Mesh Type	MESH	INTRABAND	INTERBAND	COMPOSITE	INTRABAND	INTERBAND	COMPOSITE
Non-Normal	Venus	0.3727	0.1902	0.6886	0.8320	0.5591	1.5847
	Bunny	0.3960	0.1992	0.6844	0.8033	0.5628	1.5427
	Horse	0.5615	0.2869	0.9873	1.0482	0.6943	1.9684
	Rabbit	0.4048	0.2017	0.7089	0.8996	0.6450	1.7425
	Feline	0.8277	0.2134	1.0696	1.1471	0.6285	2.0287
Normal	Venus	0.3052	0.2130	0.5741	-	-	-
	Skull	0.3381	0.2922	0.7001	-	-	-
	Dino	0.3043	0.2804	0.6672	-	-	-

Table 3. Average mutual information in bits for several non-normal and normal meshes.

Table 3 shows the average mutual information results for interband, intraband and composite dependencies for various mesh models. Since in mesh coding three different components need to be coded for each vertex position in space, the average mutual information $I_{avg} = (I_X + I_Y + I_Z)/3$ is reported instead of the mutual information for the three components individually. It is observed from Table 3 that for both normal and non-normal meshes mutual information of interband models is the least, independent of the wavelet transform employed. On the other hand mutual information for intraband models is significantly higher than for the interband models. Finally, composite models which gather the characteristics of both interband and intraband models exhibit even higher mutual information than interband or intraband models alone. Mathematically we can summarize our numerical findings as:

$$I(X;P_X) \ll I(X;\mathbf{n}_X) \ll I(X;P_X;\mathbf{n}_X)\,. \tag{26}$$

Experimental results for the mutual information based estimation of interband, intraband and composite dependencies seem to indicate that exploiting the composite dependencies should be preferred. Additionally, it is important to point out that favoring intraband over zerotree-based interband models brings along the additional benefit of resolution scalability. Specifically, by following an intraband codec design, only those wavelet subbands that are needed in order to reconstruct a target mesh resolution-level need to be encoded, while the others can be discarded. This does not hold in case of interband and composite codec designs, due to the tree-structures that span all the wavelet decomposition levels. Since composite models cannot be discarded altogether due to their highest mutual information property, a careful implementation of a composite mesh coding system needs to be carried out into order to get the benefit of both the higher compression efficiency and the resolution scalable decoding at the same time.

Finally, it is important to point out that the differences in terms of mutual information do not give any indication about the final performance differences between interband, intraband and composite coding systems. Hence, an actual development and comparison of such coding systems is needed in order to experimentally validate the conclusions of this

information-theoretic analysis of wavelet-based mesh coding designs, which is presented next.

5. Scalable mesh compression overview

In this section, we give a brief overview of the scalable mesh compression systems. Based on the design choices established earlier, we designed intraband and composite mesh coding systems which provide state-of-the-art compression performance, together with resolution as well as quality scalability of the compressed mesh.

5.1 Progressive Geometry Compression (PGC)

The first scalable wavelet-based geometry compression technique is the progressive geometry compression (PGC) codec proposed by Khodakovsky et al. in (Khodakovsky et al., 2000). PGC makes use of the well-know zero-tree coding (Shapiro, 1993) of wavelet coefficient's bitplanes in order to encode the decomposed mesh structure. Significant improvements in the compression performance against the contemporary scalable as well as non-scalable mesh coding systems were reported in (Khodakovsky et al., 2000). However, a major drawback of PGC schemes is their inability to provide resolution scalability. This is caused by the zero-tree structure which, for a given bitplane, spans all the wavelet decomposition levels. For a detailed understanding of the PGC system we refer to (Khodakovsky et al., 2000).

5.2 Scalable Intraband Mesh Compresion (SIM)

Despite of the great success of zerotree-based coding techniques in image coding, the choice of an interband codec design is not necessarily the best option in the context of scalable mesh coding. This was illustrated in Section 4 where different types of dependencies among wavelet coefficients were studied. Based on this analysis, we opt for an intraband dependency model in our codec design. As mentioned before, favoring intraband models over interband models brings along the additional benefit of resolution scalability. Specifically, by following an intraband codec design, only those wavelet subbands that are needed in order to reconstruct a target mesh resolution-level need to be encoded, while the others can be discarded.

In the designed scalable intraband mesh (SIM) compression system (Denis et al., 2010b) each resolution subband is encoded independently of the others. Similar to (Shapiro, 1993), SAQ is applied to each resolution subband to determine the significance of the wavelet coefficients with respect to a series of monotonically decreasing thresholds. Based on the significance outcome, a tree node is split into eight equal volume nodes. The resulting octree nodes may contain an unequal number of wavelet coefficients. In general, the number of coefficients in all nodes of a same tree-depth is roughly the same. This way, an octree is constructed for each resolution subband, wherein the depth of the tree (number of levels in the octree) is equal to the number of bitplanes of the subband. All magnitude bitplanes are sequentially coded using the non-significance, the significance and the refinement coding passes. For a detailed presentation of the SIM codec the interested reader is referred to (Denis et al., 2010b).

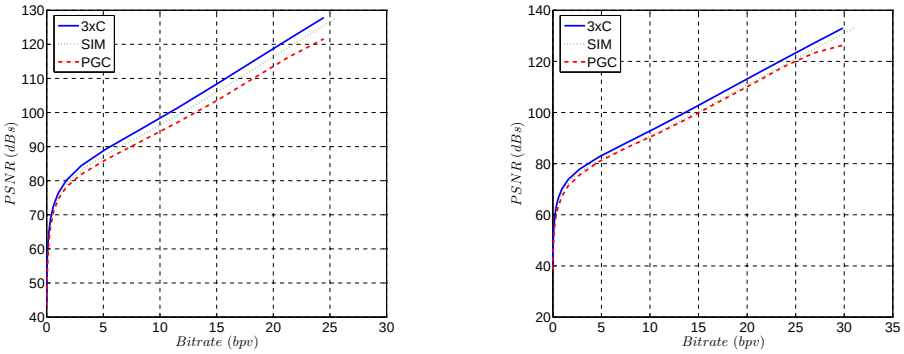
Using the octree-based bitplane coding, separate symbol streams are first generated for all bitplanes of each resolution subband. Depending on the type of scalability, i.e., resolution or quality scalability, the encoded symbol streams are entropy coded using a predefined progression order of bitplanes. For quality scalability, bitplanes of certain significance, from all resolution subbands, are first encoded before encoding the bitplanes of lower significance. However, in resolution scalability mode, all bitplanes of a lower resolution subband are progressively encoded before encoding the next higher resolution subband.

We compared the SIM codec with the PGC codec for both normal and non-normal 3D meshes. The decoded meshes are compared against the original semi-regular input meshes using the peak signal-to-noise ratio (PSNR) as the distortion metric, which is defined as:

$$PSNR = 20 \cdot \log_{10}\left(\frac{peak}{RMS}\right)(\text{dBs}),$$

where *peak* and *RMS* denote the size of the bounding box and the root mean squared error calculated on the distances between the decoded vertex positions with respect to the original ones, respectively.

Fig. 9 depicts PSNR versus bitrate (bits per semi-regular vertex) plots, evaluated for the semi-regular non-normal Venus and Bunny meshes using the Butterly transform. The results demonstrate that for both meshes, SIM yields superior performance when compared to PGC.



(a) (b)

Fig. 9. PSNR versus bitrate for non-normal mesh models in the quality scalability mode: (a) Venus, (b) Bunny . The lifted Butterly transform is employed for all three codecs.

The averaged gain in PSNR when compressing the Venus and Bunny meshes goes up to 2.22 dB and 2.35 dB, respectively. One may also notice the increasing performance difference with increasing bitrates; this indicates that the SIM coder tends to code the high frequency information more efficiently. For the spatially adaptive wavelet transform (SAWT) the compression results are reported in (Denis et al., 2010a).

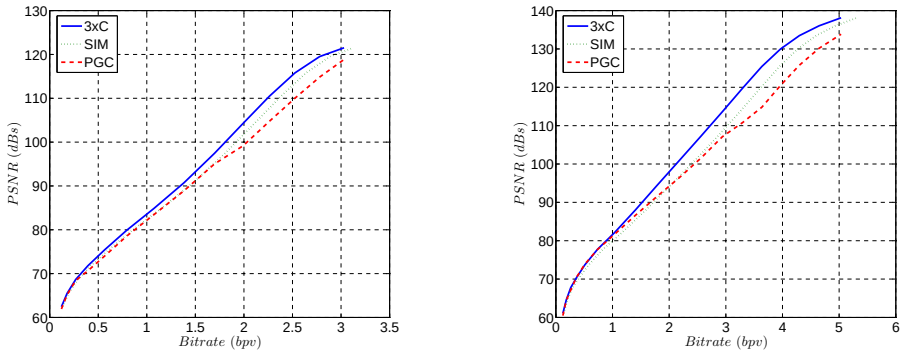


Fig. 10. PSNR versus bitrate for normal mesh models in the quality scalability mode: (a) Skull, (b) Dino. The un-lifted Butterly transform is employed for all three codecs.

Fig. 10 shows compression performance plots for two normal meshes, Skull and Dino. One notices that at low bitrates, PGC tends to compress better. However, the ability of SIM to capture and code more efficiently the high-frequency components is noticeable at high bitrates and leads to an improved performance when compared to PGC.

5.3 Composite Context-conditioned Compression (3xC)

The mutual information analysis presented earlier showed that the composite dependencies between the wavelet coefficients are by far the strongest. However, one may notice that, employing composite models may hinder, similar to interband models, the possibility of providing resolution scalability. Thus one must be careful in exploiting the parent-children dependencies within composite models. A careful observation reveals that exploiting parent-children dependencies in a causal fashion (Denis et al., 2010b) does not limit resolution scalable decoding of the compressed mesh. Following this observation, we proposed a scalable composite mesh compression system in (Denis et al., 2009), (Denis et al., 2010b). The bitplane coding modules of the SIM codec and the 3xC codec are identical. The two designs differ at the entropy coding level. In particular, for 3xC, parent coefficient based context-conditioning is employed in the entropy coding module. For context-conditioning, significant, non-significant as well as sign information is entropy coded using the designed context tables. The refinement information is encoded without context-conditioning; this is because including the parental information when entropy coding the refinement symbols does not improve compression performance. For a detailed presentation of the 3xC codec the interested reader is referred to (Denis et al., 2009).

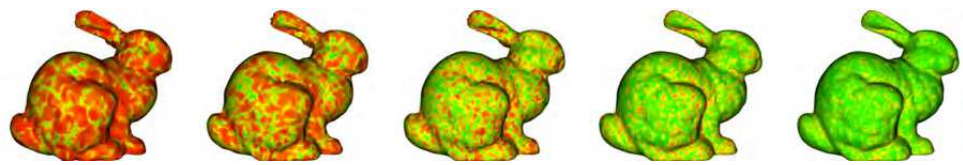
Fig. 9 also depicts the PSNR curves computed for the non-normal Venus and Bunny meshes using our implementation of the un-lifted butterfly based 3xC mesh compression system. The figure clearly demonstrates that, when dealing with non-normal meshes, 3xC systematically yields superior performance compared to PGC as well as SIM.

In the case of normal meshes (Fig. 10) our coder employs the same transform as PGC. Both codecs perform the same at very low bitrates. However, overall, 3xC yields the best compression performance. 3xC gives approximately equivalent results when compared with the intraband SIM codec for normal meshes. This is because the context-conditioning is only possible for the normal component of vector valued wavelet coefficients. Overall, it is clear that the proposed 3xC codec produces similar, and in almost all cases, superior performance compared to PGC and SIM codecs.

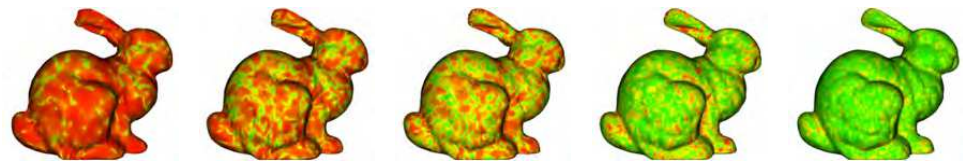
5.4 Visual comparison: PGC vs 3xC

Visual comparisons of Bunny and Skull meshes, compressed and reconstructed using 3xC at different bits per vertex (bpv), are presented in Fig. 11 and Fig. 12, respectively. The colored regions highlight the distortions introduced by lossy compression. For low-to-medium bitrates, the pure red color indicates areas where the distance between the original and decoded vertex is larger than 0.1% of the diagonal of the bounding box of the semi-regular mesh. For high bitrates, the distortion is visualized with respect to 0.02% of the diagonal. The mesh is shaded greener as the distortion lowers, with pure green indicating no distortion.

When visually comparing the compressed Bunny and Skull meshes produced by 3xC and PGC, it is very clear that 3xC yields superior performance for all bitrates. Taking the result at 0.050 bpv as an example, we observe that many areas which are shaded red for PGC are green for 3xC. At high rates, the differences between the mesh geometries may not be visually significant, yet the colors reveal that 3xC is able to approximate the original mesh much more accurately compared to the PGC system.



0.050 bpv, 51.3 dB | 0.098 bpv, 55.0 dB | 0.178 bpv, 58.7 dB | 0.314 bpv, 62.5 dB | 0.540 bpv, 66.3 dB



0.050 bpv, 47.6 dB | 0.098 bpv, 51.4 dB | 0.178 bpv, 55.4 dB | 0.314 bpv, 59.2 dB | 0.540 bpv, 63.1 dB

Fig. 11. Visual comparison of non-normal Bunny mesh using (top row) the 3xC codec and (bottom row) the PGC codec. The red color intensity reflects the distortion with respect to the uncompressed semi-regular mesh. The rate for the base mesh (i.e., M^0 - see section 2.1.2) is not included in the reported rate values.

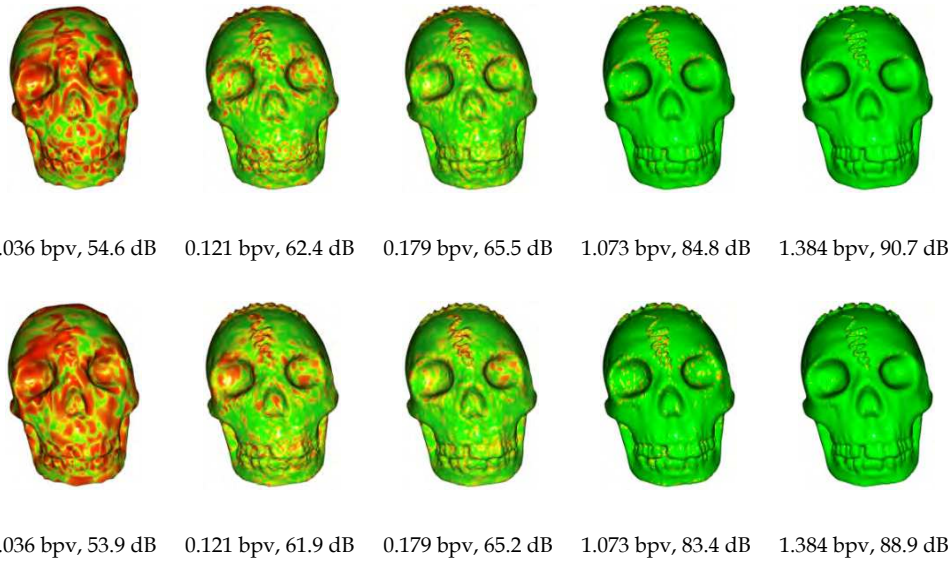


Fig. 12. Visual comparison of normal Skull mesh using (top row) the 3xC codec and (bottom row) the PGC codec. The red color intensity reflects the distortion with respect to the uncompressed semi-regular mesh. The rate for the base mesh is not included in the reported rate values.

The visual comparisons of the normal mesh Skull at different bpv are shown in Fig. 12. Though, at first glance it may appear that both codecs perform very similar, small differences are noticeable when investigating the meshes more closely. When examining the comparison at 0.036 bpv, we notice that the PGC codec preserves more details in Skull's teeth. The green shade for 3xC at rate 0.179 bpv, however, seems more pure compared to PGC for which it is rather yellowish green. We also observe that no red regions are present for 3xC at rate 1.073 bpv, whereas some are visible for PGC at the same rate.

6. Conclusions

In this book chapter, we propose a constructive methodology for the design of scalable wavelet-based mesh compression systems. Our design strategy differs from conventional designs which simply opt for reusing methods from wavelet-based image coding for the design of mesh coding systems. In particular, our methods are motivated by an information-theoretic analysis of the statistical dependencies between wavelet coefficients which shows that, intraband dependencies are systematically stronger than interband ones for both normal and non-normal meshes, and that composite models are the best. We also investigate the optimality of successive approximation quantization, commonly used in scalable compression, in the context of wavelet-based mesh compression. Using a Laplacian mixture model, it is shown that successive approximation quantization is an acceptable, but in general not an optimal solution. Anchored in these results, novel intraband and composite coding systems are presented which improve the state-of-the-art in scalable mesh compression, both in terms of scalability and compression efficiency.

7. Appendix

The output distortion D_L of a Laplacian PDF, quantized using an n level EDSQ and reconstructed using midpoint reconstruction, can be written as:

$$D_L(Q_{\xi_n,\Delta_n}) = \underbrace{\lambda\int_0^{(1-\xi_n)\Delta_n} x^2 e^{-\lambda x}\,dx}_{D_{DZ}} + \underbrace{\lambda\sum_{k=1}^{\infty}\int_{(k-\xi_n)\Delta_n}^{(k+1-\xi_n)\Delta_n}\left(x-(k+0.5-\xi_n)\Delta_n\right)^2 e^{-\lambda x}\,dx}_{D_{REST}},$$

where D_{DZ} and D_{REST} denote the distortion contributions of the deadzone and the other quantization cells, respectively. By proper substitution and letting

$$\sum_{k=1}^{\infty} e^{-\lambda\Delta_n k} = \frac{e^{-\lambda\Delta_n}}{1-e^{-\lambda\Delta_n}}, \text{ as } e^{-\lambda\Delta_n} \le 1, \tag{27}$$

the following closed-form expression for the distortion is obtained:

$$D_L(Q_{\delta_n,\Delta_n}) = \frac{2}{\lambda^2} + e^{-\lambda\Delta_n\delta_n}\left\{\left(\frac{1}{4}-\delta_n^2\right)\Delta_n^2 - \left(2\delta_n + \coth\left(\frac{\lambda\Delta_n}{2}\right)\right)\frac{\Delta_n}{\lambda}\right\}, \tag{28}$$

where $\delta_n = 1-\xi_n$.

Similarly, the output rate R_L of a Laplacian PDF, quantized using an n level EDSQ can be written as:

$$R_L(Q_{\xi_n,\Delta_n}) = \underbrace{-2\left(\frac{\lambda}{2}\int_0^{(1-\xi_n)\Delta_n} e^{-\lambda x}\,dx\right)\log_2 2\left(\frac{\lambda}{2}\int_0^{(1-\xi_n)\Delta_n} e^{-\lambda x}\,dx\right)}_{R_{DZ}}\ldots$$
$$\ldots\underbrace{-2\sum_{k=1}^{\infty}\left(\frac{\lambda}{2}\int_{(k-\xi_n)\Delta_n}^{(k+1-\xi_n)\Delta_n} e^{-\lambda x}\,dx\right)\log_2\left(\frac{\lambda}{2}\int_{(k-\xi_n)\Delta_n}^{(k+1-\xi_n)\Delta_n} e^{-\lambda x}\,dx\right)}_{R_{REST}}.$$

Again making use of the summation reduction identity of (27) along with the identity

$$\sum_{k=1}^{\infty} e^{-\lambda\Delta_n k}\log_2\left(e^{-\lambda\Delta_n k}\right) = \log_2\left(e^{-\lambda\Delta_n}\right)\sum_{k=1}^{\infty} k\left(e^{-\lambda\Delta_n}\right)^k = \frac{\log_2\left(e^{-\lambda\Delta_n}\right)e^{-\lambda\Delta_n}}{\left(1-e^{-\lambda\Delta_n}\right)^2},$$

the expression for the rate can be reduced to the following closed-form:

$$R_L(Q_{\delta_n,\Delta_n}) = c_\delta\log_2\left(\frac{2d_{\delta_n}}{d_1 c_1^{1/d_1} e^{\lambda\Delta_n(1-\delta_n)} d_{\delta_n}^{1/c_{\delta_n}}}\right), \tag{29}$$

where $c_{\delta_n} = e^{-\lambda\Delta_n\delta_n}$ (hence $c_1 = e^{-\lambda\Delta_n}$) and $d_{\delta_n} = 1-c_{\delta_n}$ (hence $d_1 = 1-c_1$).

8. Acknowledgements

The authors would like to thank Cyberware, Headus, The Scripps Research Institute, Washington University, and Stanford University for providing 3D models. The authors are particularly grateful to Igor Guskov for providing them with the normal meshes, and to Andrei Khodakovsky and Peter Schröder for providing the PGC software.

This research was supported by the Agency for Innovation by Science and Technology (IWT) - Flanders (OptiMMa project) and the Fund for Scientific Research – Flanders (postdoctoral mandate Peter Schelkens and project G014610N).

9. References

Catmull, E. & Clark, J. (1978). Recursively Generated B-Spline Surfaces on Arbitrary Topological Surfaces. *Computer-Aided Design*. Vol. 10, No. 6, (November 1978), pp. 350-355, ISBN 1-58113-052-X.

Chang, C.-L. & Girod, B. (2006). Direction-Adaptive Discrete Wavelet Transform Via Directional Lifting and Bandeletization. Proc. *IEEE International Conference on Image Processing*, pp. 1149-1152, Atlanta, GA, USA.

Cover, T. M. & Thomas, J. A. (1991). *Elements of Information Theory*. Wiley-Interscience, ISBN 0-471-24195-4, New York, USA.

Darbellay, G. A. & Vajda, I. (1999). Estimation of the Information by an Adaptive Partitioning of the Observation Space. *IEEE Transactions on Information Theory*. Vol. 45, No. 4, (May 1999), pp. 1315-1321, ISSN 0018-9448.

Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data Via the Em Algorithm. *Journal of The Royal Statistical Society, Series B*. Vol. 39, No. 1, (May 1977), pp. 1-38.

Denis, L., Ruxandra, F., Munteanu, A. & Schelkens, P. (2010a). Spatially Adaptive Bases in Wavelet-Based Coding of Semi-Regular Meshes. *Proceedings of the SPIE*, pp. 772310-772310-8, Brussels, Belgium.

Denis, L., Satti, S. M., Munteanu, A., Cornelis, J. & Schekens, P. (2010b). Scalable Intraband and Composite Coding of Semi-Regular Meshes. *IEEE Transactions on Multimedia*. Vol. 12, No. 8, (December 2010), pp. 773-789, ISSN 1520-9210.

Denis, L., Satti, S. M., Munteanu, A., Cornelis, J. & Schelkens, P. (2009). Context-Conditioned Composite Coding of 3D Meshes Based on Wavelets on Surfaces. *IEEE International Conference on Image Processing*, pp. 3509-3512, ISSN 1522-4880, Cairo, Egypt, November 2009.

Dyn, N., Levin, D. & Gregory, J. A. (1990). A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control. *ACM Transactions on Graphics*. Vol. 9, No. 2, (April 1990), pp. 160-169, ISSN 0730-0301.

Eck, M., Derose, T., Duchamp, T., Hoppe, H., Lounsbery, M. & Stuetzle, W. (1995). Multiresolution Analysis of Arbitrary Meshes. *ACM SIGGRAPH, Proceedings of the 22rd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 173-182 ISBN 0-89791-701-4, Los Angeles, California, USA, August 6-11, 1995.

Eck, M. & Hoppe, H. (1996). Automatic Reconstruction of B-Spline Surfaces of Arbitrary Topological Type. *ACM SIGGRAPH, Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 325–334, ISBN 0-89791-746-4.

Fraysse, A., Pesquet-Popescu, J. C. (2008) Rate-distortion Results on Generalized Gaussian Distributions, *IEEE International Conference on Acoustic, Speech Signal Processing,* pp. 3753-3756, Las Vegas, NV, USA, March 30-April 04, 2008.

Hsiang, S.-T. & Woods, J. W. (2000). Embedded Image Coding Using Zeroblocks of Subband/Wavelet Coefficients and Context Modeling. *IEEE International Symposium on Circuits and Systems*, pp. 662-665, Geneva, CH, May 28-31.

Khodakovsky, A., Schröder, P. & Sweldens, W. (2000). Progressive Geometry Compression. *ACM SIGGRAPH, 27th International Conference on Computer Graphics and Interactive Techniques*, pp. 271-278, ISBN 1-58113-208-5, New Orleans, Lousiana, USA, July 23-28, 2000.

Li, J. & Kuo, C. C. J. (1998). Progressive Coding of 3-D Graphic Models. *Proceedings of the IEEE* Vol. 86, No. 6, pp. 1052-1063, ISSN 0018-9219.

Liu, J. & Moulin, P. (2000). Analysis of Interscale and Intrascale Dependencies between Image Wavelet Coefficients. *International Conference on Image Processing*, pp. 669-671, Vancouver, Canada, September 11-13, 2000.

Loop, C., Schaefer, S., Ni, T. & Castaño, I. (2009). Approximating Subdivision Surfaces with Gregory Patches for Hardware Tessellation. *ACM Transactions on Graphic, SIGGRAPH Asia*. Vol. 28, No. 5, (December 2009), pp. 151:1 - 151:9, ISSN 0730-0301.

Lounsbery, M., Derose, T. D. & Warren, J. (1997). Multiresolution Analysis for Surfaces of Arbitrary Topological Type. *ACM Transactions on Graphics*. Vol. 16, No. 1, (January 1997), pp. 34-73.

Mallat, S. G. (1989). A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 11, No. 7, (July 1989), pp. 674-693, ISSN 0162-8828.

Munteanu, A., Cornelis, J. & Cristea, P. (1999a). Wavelet-Based Lossless Compression of Coronary Angiographic Images. *IEEE Transactions on Medical Imaging*. Vol. 18, No. 3, (March 1999), pp. 272-281.

Munteanu, A., Cornelis, J., Van der Auwera, G. & Cristea, P. (1999b). Wavelet Image Compression - the Quadtree Coding Approach. *IEEE Transactions on Information Technology in Biomedicine*. Vol. 3, No. 3, (September 1999), pp. 176-185, ISSN 1089-7771.

Pajarola, R. & Rossignac, J. (2000). Compressed Progressive Meshes. *IEEE Transactions on Visualization and Computer Graphics*. Vol. 6, No. 1, (January-March 2000), pp. 79-93, ISSN 1077-2626.

Ronfard, R. & Rossignac, J. (1996). Full-Range Approximation of Triangulated Polyhedra. *Proceeding of Eurographics, Computer Graphics Forum*. Vol. 15, No. 3, (August 1996), pp. 67-76, ISSN 0167-7055.

Rossignac, J. & Borrel, P. (1993). *Multi-Resolution 3d Approximation for Rendering Complex Scenes*. Springer-Verlag, ISBN 0387565299.

Schröder, P. & Sweldens, W. (1995). Spherical Wavelets: Efficiently Representing Functions on the Sphere. *ACM SIGGRAPH, Proceedings of 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 161-172, ISBN 0-89791-701-4, Los Angeles, California, USA, August 6-11.

Shapiro, J. M. (1993). Embedded Image Coding Using Zerotrees of Wavelet Coefficients. *IEEE Transactions on Signal Processing*. Vol. 41, No. 12, (1993), pp. 3445-3462.

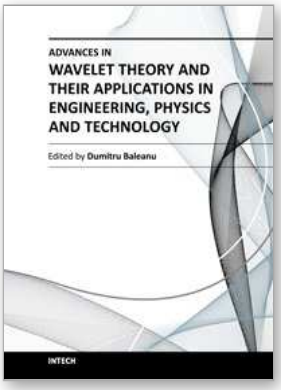
Soucy, M. & Laurendeau, D. (1996). Multiresolution Surface Modeling Based on Hierarchical Triangulation. *Computer Vision and Image Understanding*. Vol. 63, No. 1, (January 1996), pp. 1-14, ISSN 10773142.

Taubin, G., Guéziec, A., Horn, W., William, A. & Lazarus, F. (1998). Progressive Forest Split Compression. *ACM SIGGRAPH, Proceedings of 25th International Conference on Computer Graphics and Interactive Techniques*, pp. 123-132, ISBN 0-89791-999-8, Orlando, Florida, USA, July 19-24, 1998.

Taubman, D. (2000). High Performance Scalable Image Compression with EBCOT. *IEEE Transactions on Image Processing*. Vol. 9, No. 7, (July 2000), pp. 1158-1170, ISSN 1057-7149.

Taubman, D. & Marcelin, M. (2001). *Jpeg2000: Image Compression Fundamentals, Standards and Practice*. Springer, ISBN 978-0792375197.

Wu, X. (1997). High-Order Context Modeling and Embedded Conditional Entropy Coding of Wavelet Coefficients for Image Compression. *31st Asilomar Conference on Signals, Systems and Computers*, pp. 1378-1382, Pacific Grove, CA, November 2-5, 1997.



Advances in Wavelet Theory and Their Applications in Engineering, Physics and Technology

Edited by Dr. Dumitru Baleanu

ISBN 978-953-51-0494-0
Hard cover, 634 pages
Publisher InTech
Published online 04, April, 2012
Published in print edition April, 2012

The use of the wavelet transform to analyze the behaviour of the complex systems from various fields started to be widely recognized and applied successfully during the last few decades. In this book some advances in wavelet theory and their applications in engineering, physics and technology are presented. The applications were carefully selected and grouped in five main sections - Signal Processing, Electrical Systems, Fault Diagnosis and Monitoring, Image Processing and Applications in Engineering. One of the key features of this book is that the wavelet concepts have been described from a point of view that is familiar to researchers from various branches of science and engineering. The content of the book is accessible to a large number of readers.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821