COMBINED QUADRIC ERROR METRIC AND LIFTING SCHEME MULTIVARIATE MODEL SIMPLIFICATION

Teodor Cioacă[1], Bogdan Dumitrescu[2], Mihai-Sorin Stupariu[3]

The quadric error metric incremental simplification process involves two stages: cost computation and vertex estimation. In this work, we replace the latter through the use of a lifting scheme prediction-update stage. Since the initial simplification algorithm implies the fusion of neighboring vertices and their associated matrices, we propose the inclusion of a similar mechanism in the update stage of the lifting scheme. For this purpose, we explore two possible choices for the redistribution weights of the quadric error matrices. The quality of the data downsampled in this fashion is assessed by tracking the evolution of the normalized root mean squared error.

Keywords: multivariate lifting scheme, quadric error metric, predict-update

1. Introduction

A common issue arising in the interpretation process of high density digitized models is the barrier data richness itself poses. Typical laser scanners can produce highly redundant point clouds where only the high frequency details are directly observable. One solution to this problem is the so-called *incremental simplification* [1]. In this approach, the vertices of the model are sequentially removed according to a cost heuristic designed to minimize the approximation error. However, the resulting, coarser approximations do not compensate for the loss of information in a way that guarantees the preservation of any properties of the initial model. *Wavelet multiresolution analysis* (WMRA) can be viewed as an alternative simplification method that is constructed such that it preserves the mean of the signal associated with the input

[1]Department of Automatic Control and Systems Engineering, Politehnica University of Bucharest, Splaiul Independenţei 313, 060042 Bucharest, Romania, e-mail: t.cioaca@gmail.com

[2]Department of Automatic Control and Systems Engineering, Politehnica University of Bucharest, Splaiul Independenţei 313, 060042 Bucharest, Romania, e-mail: bogdan.dumitrescu@acse.pub

[3]Department of Computer Science, University of Bucharest, Strada Academiei 14, Bucharest, Romania email: stupariu@fmi.unibuc.ro

model. The *second generation wavelet transform*, introduced by Sweldens [2], can be constructed to meet this goal through the *lifting scheme* design.

As a follow-up to the work by Cioaca et al. [3, 4], we aim to combine the feature identification capabilities of the *quadric error metric*, introduced by Garland and Heckbert [5] with the antialiasing and variance reduction properties of a lifting scheme design, proposed by Jansen in [6]. Chronologically, wavelet analysis was constructed starting from mesh subdivision operators. In this sense, Valette and Prost [7] generalized a face subdivision scheme for triangle meshes, while Bertram [8] conceived a hybrid, edge-collapse and predictor-corrector wavelet downsampling method. Zhao and Sun [10] proposed a matrix-valued subdivision as an extension of the scalar-valued subdivision. In this work, the requirement to use an inverse subdivision operator is no longer present, as we make use of a lifting design capable of downsampling arbitrary graphs with manifold connectivity. We refer the interested reader to the survey of Shuman et al. [11] where fundamental signal processing operations are generalized to the multidimensional graph setting in both vertex and spectral domains.

The remainder of this work is organized as follows. In section 2 we describe the construction of the generalized quadric error metric and its role in establishing an even-odd node partition. Then, a different mathematical derivation of the one-dimensional lifting scheme design proposed by Jansen is presented in section 3. The graph extension of this scheme follows in section 4, while numerical experiments and conclusions are provided in sections 5 and 6.

2. Generalized quadric error metric

We briefly review the construction of the generalized quadric error metric, first detailed by Garland and Heckbert [5]. The purpose of this representation is to facilitate computing the squared distances from any point to the support plane of a triangle, regardless of the dimensionality of the space where these entities are embedded in. That is, if $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \in \mathbb{R}^n$ are column vectors corresponding to the vertices of a triangle and $\mathbf{p} \in \mathbb{R}^n$ is an arbitrary point, we are interested in how the squared distance from $\mathbf{p}$ to the $\Delta(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ triangle can be expressed. The support plane of $\Delta(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ is uniquely determined by two linearly independent vectors. Moreover, one can attempt and construct an orthonormal frame such that two of its basis vectors are contained within this plane. This process only implies applying the first two steps of the Gram-Schmidt algorithm, i.e.:

$$\hat{\mathbf{e}}_1 = \frac{\mathbf{v}_2 - \mathbf{v}_1}{\|\mathbf{v}_2 - \mathbf{v}_1\|}, \tag{1}$$

and

$$\hat{\mathbf{e}}_2 = \frac{\mathbf{v}_3 - \mathbf{v}_1 - ((\mathbf{v}_3 - \mathbf{v}_1)^\intercal \hat{\mathbf{e}}_1)\, \hat{\mathbf{e}}_1}{\|\mathbf{v}_3 - \mathbf{v}_1 - ((\mathbf{v}_3 - \mathbf{v}_1)^\intercal \hat{\mathbf{e}}_1)\, \hat{\mathbf{e}}_1\|}. \tag{2}$$

The Gram-Schmidt algorithm can be used to completely recover the remaining $n-2$ basis vectors. Assuming only that one can uniquely identify these vectors, the expression of $\mathbf{p}-\mathbf{v}_1$ in this basis is written as:

$$\mathbf{p}-\mathbf{v}_1=\sum_{i=1}^{n}\left((\mathbf{p}-\mathbf{v}_1)^\intercal\hat{\mathbf{e}}_i\right)\hat{\mathbf{e}}_i. \tag{3}$$

Consequently,

$$\|\mathbf{p}-\mathbf{v}_1\|^2=\sum_{i=1}^{n}\left((\mathbf{p}-\mathbf{v}_1)^\intercal\hat{\mathbf{e}}_i\right)^2. \tag{4}$$

Due to the way the $\hat{\mathbf{e}}_i$ vectors were constructed, equation (4) reveals the squared distance from $\mathbf{p}$ to the support plane of $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$. The sum of the squared lengths of all components that $\mathbf{p}-\mathbf{v}_1$ has when projected onto the $n-2$ remaining basis vectors is exactly the expression of this squared distance, i.e. the left-hand side of the following equation:

$$\|\mathbf{p}-\mathbf{v}_1\|^2-\left((\mathbf{p}-\mathbf{v}_1)^\intercal\hat{\mathbf{e}}_1\right)^2-\left((\mathbf{p}-\mathbf{v}_1)^\intercal\hat{\mathbf{e}}_2\right)^2=\sum_{i=3}^{n}\left((\mathbf{p}-\mathbf{v}_1)^\intercal\hat{\mathbf{e}}_i\right)^2. \tag{5}$$

The squared distance expressed in equation (5) can be rewritten as:

$$d(\mathbf{p},\Delta(\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3))^2=\begin{pmatrix}\mathbf{p}\\1\end{pmatrix}^\intercal\begin{bmatrix}\mathbf{A}&\mathbf{b}\\\mathbf{b}^\intercal&c\end{bmatrix}\begin{pmatrix}\mathbf{p}\\1\end{pmatrix}, \tag{6}$$

where

$$\mathbf{A} = \mathbf{I}-\hat{\mathbf{e}}_1\hat{\mathbf{e}}_1^\intercal-\hat{\mathbf{e}}_2\hat{\mathbf{e}}_2^\intercal, \tag{7}$$

$$\mathbf{b} = (\mathbf{v}_1^\intercal\hat{\mathbf{e}}_1)\hat{\mathbf{e}}_1+(\mathbf{v}_1^\intercal\hat{\mathbf{e}}_2)\hat{\mathbf{e}}_2-\mathbf{v}_1, \tag{8}$$

$$c = \mathbf{v}_1^\intercal\mathbf{v}_1-(\mathbf{v}_1^\intercal\hat{\mathbf{e}}_1)^2-(\mathbf{v}_1^\intercal\hat{\mathbf{e}}_2)^2. \tag{9}$$

By denoting $\mathbf{Q}(\Delta(\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3))=\begin{bmatrix}\mathbf{A}&\mathbf{b}\\\mathbf{b}^\intercal&c\end{bmatrix}$, one can algebraically express the sum of squared distances from $\mathbf{p}$ to the support planes of a set of triangles, $(\Delta_i(\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3))_{i=\overline{1,N}}$, as:

$$\sum_{i=\overline{1,N}}d(\mathbf{p},\Delta_i(\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3))^2=\begin{pmatrix}\mathbf{p}\\1\end{pmatrix}^\intercal\left\{\sum_{i=\overline{1,N}}\mathbf{Q}\left(\Delta_i(\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3)\right)\right\}\begin{pmatrix}\mathbf{p}\\1\end{pmatrix}. \tag{10}$$

In the original incremental simplification algorithm from [5], the set of faces in the one-ring neighborhood of each vertex is used to compute an associated matrix:

$$\mathbf{Q}(\mathbf{v})=\sum_{\Delta_k\in\mathcal{N}_f^1(\mathbf{v})}\mathbf{Q}(\Delta_k), \tag{11}$$

where $\mathcal{N}_f^1(\mathbf{v})$ represents the set of all triangles incident at $\mathbf{v}$. Whenever two vertices, $\mathbf{v}_a$ and $\mathbf{v}_b$, are fused into a new vertex, $\mathbf{w}$, as a result of a collapse

operation, the local geometric information that was characterized by the neighborhoods of these vertices is preserved by setting $\mathbf{Q}(\mathbf{w}) \leftarrow \mathbf{Q}(\mathbf{v}_a) + \mathbf{Q}(\mathbf{v}_b)$.

The matrix terms from equation (11) describe quadrics in the sense that all isosurfaces obtained from varying point $\mathbf{p}$ in equation (10) are quadrics. The term *quadric error metric* is thus justified since these matrices offer a means of estimating an error measure from an arbitrary position $\mathbf{p}$ to a local patch around any vertex $\mathbf{v}$. As described in [4], this metric also allows for the introduction of a cost function associated to each vertex:

$$\texttt{cost}(\mathbf{v}) = \begin{pmatrix} \mathbf{v} \\ 1 \end{pmatrix}^{\intercal} \left(\sum_{\mathbf{v}_i \in \mathcal{N}_v^1(\mathbf{v})} \mathbf{Q}(\mathbf{v}_i) \right) \begin{pmatrix} \mathbf{v} \\ 1 \end{pmatrix}, \tag{12}$$

where $\mathcal{N}_v^1(\mathbf{v})$ denotes the direct neighbors of $\mathbf{v}$, or the *one-ring* vertex set.

We now consider the case of a triangular manifold $M = (V, E, F)$, where V is the set of vertices or nodes, E is the set of edges and F the set of faces of this manifold. Through the introduction of the cost function in equation (12), the elements of V can be ordered ascendingly according to their importance. Next, a lazy wavelet partitioning scheme, detailed in [4], is employed to label vertices as either *even* or *odd*. This process is straightforward and requires starting with a set of unlabelled vertices. Initially, this set contains all vertices. Then, in an iterative manner, the least important unlabelled vertex is selected and tagged as *odd*, while all of its one-ring neighbors are tagged as *even*. This process is repeated until all vertices are labelled, i.e. $V = V_O \cup V_E$, where V_O and V_E are the resulted odd and even vertex subsets.

Removing the V_O subset leads to a direct simplification algorithm that heuristically preserves the more salient features of the initial model assuming the holes left after the removal of the odd samples are filled in order to preserve manifold connectivity. If one instead wishes to interpret the entire mesh or graph as a signal, then some initial properties of this signal are lost if no compensation mechanism is used. A common example is to preserve the average of the signal, which can be intuitively interpreted as a means of redistributing the lost information amongst the neighbors of a removed graph node.

3. **Lifting scheme**

In this section we discuss in more detail our proposition for extending the lifting scheme design to the graph set-up. For a more in-depth analysis of the theory and applications of the lifting scheme we refer the reader to [9].

Introducing wavelet constructs on graphs requires first reviewing the theory in the one-dimensional case. In general, wavelets are the building blocks of hierarchical function space approximations, i.e. $L^2(\mathbb{R}) \succ \ldots \succ V_n \succ V_{n-1} \succ \ldots \succ V_1 \succ V_0 \succ \ldots \succ \{0\}$. Let V_j be a function space approximation at level j. We are interested in examining the connections between two consecutive approximations, V_{j+1} and V_j. Let $\Phi_j = \begin{bmatrix} \ldots & \phi_{j,k} & \ldots \end{bmatrix}$ be the row vector of

basis functions that generate V_j. Then a function $f_{j+1} \in V_{j+1}$ cannot be represented by using only the basis functions in V_j, but it can be expressed as a combination of the basis functions of the V_{j+1} space, that is

$$f_{j+1} = \Phi_{j+1}\mathbf{s}_{j+1}, \tag{13}$$

where $\mathbf{s}_{j+1}$ is an infinite column vector of *scaling coefficients*. The complementary basis or *wavelet functions*, $\psi_{j,k}$, that generate the orthogonal subspace, W_j, with $V_{j+1} = V_j \oplus W_j$, correspond to the lost details. One can hence write

$$f_{j+1} = \Phi_{j+1}\mathbf{s}_{j+1} = \Phi_j\mathbf{s}_j + \Psi_j\mathbf{w}_j, \tag{14}$$

where $\Psi_j = \begin{bmatrix}\dots\psi_{j,k}\dots\end{bmatrix}$ and $\mathbf{w}_j$ is the infinite column vector of *wavelet coefficients*. Equation (14) hints at the intent behind Swelden's [2] second generation wavelet: "building blocks that can decorrelate data".

Having introduced these function space entities, we can review the mechanism behind the lifting scheme. The first operation involved in a lifting step has already been discussed in section 2 in the context of multivariate graph nodes. The same reasoning can be applied to the set of scaling coefficients at level $j+1$. The vector of scaling coefficients can then be reindexed such that

$$\mathbf{s}_{j+1} = \begin{bmatrix}\mathbf{s}_{j+1,o} \\ \mathbf{s}_{j+1,e}\end{bmatrix}, \tag{15}$$

where the o and e subscripts stand for *odd* and *even* coefficients, respectively. At this point it must be noted that the even-odd splitting of the scaling coefficients is assumed to be the result of an operation suitable for one-dimensional signals. Although one can write equivalent 2-dimensional *quadratic* error metric matrix to those in equation (12), for the remainder of this section we can assume the splitting to have already been performed using a known criterion. Besides this partitioning operation, the other two that make up the lifting scheme are the *predict*,

$$\mathbf{w}_j = \mathbf{s}_{j+1,o} - \mathbf{P}_j\mathbf{s}_{j+1,e}, \tag{16}$$

and the *update*

$$\mathbf{s}_j = \mathbf{s}_{j+1,e} + \mathbf{U}_j\mathbf{w}_j, \tag{17}$$

with $\mathbf{P}_j \in \mathbb{R}^{n_o \times n_e}$ being a sparse prediction matrix, $\mathbf{U}_j \in \mathbb{R}^{n_e \times n_o}$ being a sparse update matrix and n_o and n_e being the number of odd and even coefficients, respectively. The prediction stage simply exploits the signal redundancy by assuming that the odd coefficients can be estimated as linear combinations of their spatial neighbors. If only the even coefficients are used to approximate the functions in V_{j+1}, then the lost details are compensated for by redistributing them among these remaining coefficients via the update matrix.

We propose investigating the relation between the scaling and wavelet basis functions from two consecutive levels are related. Rewriting equation (14) in matrix form we obtain

$$\begin{bmatrix}\Phi_{j+1,o} & \Phi_{j+1,e}\end{bmatrix}\begin{bmatrix}\mathbf{s}_{j+1,o} \\ \mathbf{s}_{j+1,e}\end{bmatrix} = \begin{bmatrix}\Psi_j & \Phi_j\end{bmatrix}\begin{bmatrix}\mathbf{w}_j \\ \mathbf{s}_j\end{bmatrix}, \tag{18}$$

and, expanding the scaling and wavelet coefficients on the right-hand side using equations (16) and (17), we arrive to

$$\begin{bmatrix}\Phi_{j+1,o} & \Phi_{j+1,e}\end{bmatrix}\begin{bmatrix}\mathbf{s}_{j+1,o}\\ \mathbf{s}_{j+1,e}\end{bmatrix} = \begin{bmatrix}\Psi_j & \Phi_j\end{bmatrix}\begin{bmatrix}\mathbf{s}_{j+1,o} - \mathbf{P}_j\mathbf{s}_{j+1,e}\\ (\mathbf{I} - \mathbf{U}_j\mathbf{P}_j)\mathbf{s}_{j+1,e} + \mathbf{U}_j\mathbf{s}_{j+1,o}\end{bmatrix}. \quad (19)$$

Equation (19) must hold for any combination of the level $j+1$ lifting coefficients. Hence, let $\mathbf{s}_{j+1,e} = \delta_k$, i.e. the Kronecker vector at index k, with $k \in \overline{1, n_e}$, i.e. $\delta_{k,i} = 0, \forall k \neq i$ and $\delta_{k,k} = 1$. Also, let $\mathbf{s}_{j+1,o} = \mathbf{0}$. Evaluating both sides of equation (19), we arrive to

$$\Phi_{j+1,e}\delta_k = -\Psi_j\mathbf{P}_j\delta_k + \Phi_j\delta_k - \Phi_j\mathbf{U}_j\mathbf{P}_j\delta_k, \quad (20)$$

or, since this equation holds for any $k \in \overline{1, n_e}$, a more direct formulation can be written as

$$\Phi_{j+1,e} = -\Psi_j\mathbf{P}_j + \Phi_j - \Phi_j\mathbf{U}_j\mathbf{P}_j. \quad (21)$$

By setting $\mathbf{s}_{j+1,e} = \mathbf{0}$ and $\mathbf{s}_{j+1,o} = \delta_k$, with $k \in \overline{1, n_o}$, we find that

$$\Phi_{j+1,o}\delta_k = \Psi_j\delta_k + \Phi_j\mathbf{U}_j\delta_k, \quad (22)$$

or

$$\Phi_{j+1,o} = \Psi_j + \Phi_j\mathbf{U}_j. \quad (23)$$

Right-multiplying both sides of equation (23) by $\mathbf{P}_j$ and adding the result to equation (21) we obtain:

$$\Phi_j = \Phi_{j+1,e} + \Phi_{j+1,o}\mathbf{P}_j, \quad (24)$$

and then

$$\Psi_j = \Phi_{j+1,o}(\mathbf{I} - \mathbf{P}_j\mathbf{U}_j) - \Phi_{j+1,e}\mathbf{U}_j. \quad (25)$$

Let $\varsigma_j = \int_{-\infty}^{\infty}\Phi_j^\intercal(t)dt$. Then integrating equation (24) yields:

$$\varsigma_j = \varsigma_{j+1,e} + \mathbf{P}_j^\intercal\varsigma_{j+1,o}. \quad (26)$$

Since the integral of the wavelet functions is zero, integrating equation (25) leads to:

$$\mathbf{0} = \varsigma_{j+1,o} - \mathbf{U}_j^\intercal\left(\mathbf{P}_j^\intercal\varsigma_{j+1,o} + \varsigma_{j+1,e}\right) = \varsigma_{j+1,o} - \mathbf{U}_j^\intercal\varsigma_j. \quad (27)$$

The column vectors of $\mathbf{U}_j$ can be retrieved in a one-by-one fashion from equation (27). If $\mathbf{u}_{j_k}$ is the kth column vector, then

$$\varsigma_{j+1,o_k} = \mathbf{u}_{j_k}^\intercal\varsigma_j. \quad (28)$$

Jansen [6] suggests choosing the minimum norm solution of this equation as it leads to increased numerical stability, as experimentally demonstrated in [3]. Thus

$$\mathbf{u}_{j_k} = \varsigma_{j+1,o_k}\frac{\varsigma_j}{\|\varsigma_j\|^2}, \quad (29)$$

a solution that can be easily derived using any suitable optimization method such as Lagrange multipliers.

4. **Wavelets on graphs**

The one-dimensional lifting scheme can be extended to functions defined on manifolds. Let M be a 2-dimensional manifold and $F : M \to \mathbb{R}^d$ a multivariate function defined over this manifold. Since $F = (f_1, \ldots, f_d)$ can be viewed as a tuple of d individual square-integrable functions, $f_i : M \to \mathbb{R}$, then each of these functions can be represented using a set of basis functions also defined on the manifold. One can generalize equation (13) as

$$F_{j+1} = \Phi_{j+1}\mathbf{S}_{j+1}, \tag{30}$$

is a matrix whose columns are the $\mathbf{s}_{j+1_i}$ scaling coefficient vectors of $f_{i_{j+1}}$. The theory from the one-dimensional case does not require further changes except for working with scaling and wavelet matrices.

As described in section 2, the generalized quadric error metric provides an efficient mechanism for splitting the mesh vertices into odd and even subsets. The prediction stage only involves predicting the odd nodes from their one-ring neighbors. For most purposes, as detailed in [3], the Fujiwara Laplacian [12] is an efficient prediction operator that delivers low approximation errors. Hence, if $\mathbf{v}_{j+1,o}$ is an odd vertex, its corresponding wavelet coefficient vector is retrieved from equation (16) as

$$\mathbf{w}_{j,\mathbf{v}_{j+1,o}} = \mathbf{s}_{j+1,\mathbf{v}_{j+1,o}} - \sum_{\mathbf{v}_{j+1,e_i} \in \mathcal{N}^1(\mathbf{v}_{j+1,o})} p_{j,\mathbf{v_{j+1,o}}}(\mathbf{v}_{j+1,e_i})\mathbf{s}_{j+1,\mathbf{v}_{j+1,e_i}}, \tag{31}$$

where the prediction weights correspond to the Fujiwara Laplacian operator, i.e.

$$p_{j,\mathbf{v_{j+1,o}}}(\mathbf{v}_{j+1,e_i}) = \frac{\|\mathbf{s}_{j+1,\mathbf{v}_{j+1,o}} - \mathbf{s}_{j+1,\mathbf{v}_{j+1,e_i}}\|^{-1}}{\sum_{\mathbf{v}_{j+1,e_k} \in \mathcal{N}^1(\mathbf{v}_{j+1,o})} \|\mathbf{s}_{j+1,\mathbf{v}_{j+1,o}} - \mathbf{s}_{j+1,\mathbf{v}_{j+1,e_k}}\|^{-1}}. \tag{32}$$

Since the purpose of the wavelet decomposition is to construct hierarchical approximations, $V_k(M)$, of the square-integrable function space $L^2(M)$, one needs to find a set of generating scaling functions for $V_k(M)$. One option is to choose $\varphi_{i\in\overline{1,k}}$ such that $\int_M \varphi_i = 1$. Computing manifold integrals in equations (26) and (27), we arrive to the same formula fo the update weights. Thus, for any $\mathbf{v}_{j+1,e_i} \in \mathcal{N}^1(\mathbf{v}_{j+1,o})$, equation (17) is equivalent to

$$\mathbf{s}_{\mathbf{v}_{j,e_i}} = \mathbf{s}_{\mathbf{v}_{j+1,e_i}} + \sum_{\mathbf{v}_{j+1,o_l} \in \mathcal{N}^1(\mathbf{v}_{j+1,e_i}) \cap V_{O,j+1}} u_{j,\mathbf{v}_{j+1,e_i}}(\mathbf{v}_{j+1,o_l})\mathbf{w}_{j,\mathbf{v}_{j+1,o_l}}. \tag{33}$$

So far, the details that are contained in the removed odd samples are stored in the wavelet coefficients and the remaining even samples are updated to preserve the mean of the original signal. Nevertheless, the question concerning the fate of the information contained within the quadric error matrices of the odd samples is still unanswered. The immediate choice is distributing the matrix corresponding to an odd sample, $\mathbf{Q}(\mathbf{v}_{j+1,o})$, among its even neighbors.

Thus, each even matrix is updated as described by the following equation

$$\mathbf{Q}(\mathbf{v}_{j,e_i}) = \mathbf{Q}(\mathbf{v}_{j+1,e_i}) + \sum_{\mathbf{v}_{j+1,o_l} \in \mathcal{N}^1(\mathbf{v}_{j+1,e_i}) \cap V_{O,j+1}} \omega_{j,\mathbf{v}_{j+1,e_i}}(\mathbf{v}_{j+1,o_l}) \mathbf{Q}(\mathbf{v}_{j+1,o_l}), \quad (34)$$

where $\omega_{j,\mathbf{v}_{j+1,e_i}}(\mathbf{v}_{j+1,o_l})$ is the redistribution weight describing the influence of $\mathbf{v}_{j+1,o_l}$ on its even neighbor, $\mathbf{v}_{j+1,e_i}$. We propose two alternative choices for these weights:

- using inverse distance weights (IDW) [13], i.e.

$$\omega_{j,\mathbf{v}_{j+1,e_i}}(\mathbf{v}_{j+1,o_l}) = \frac{\|\mathbf{s}_{j+1,e_i} - \mathbf{s}_{j+1,o_l}\|^p}{\sum_{\mathbf{v}_{j+1,e_k} \in \mathcal{N}^1(\mathbf{v}_{j+1,o_l}) \cap V_{E,j+1}} \|\mathbf{s}_{j+1,e_k} - \mathbf{s}_{j+1,o_l}\|^p}, \quad (35)$$

  where p is a negative integer exponent. For $p = -1$, the redistribution weights coincide with the prediction weights as defined in equation (32). Thus the update equation (34) for the even quadric error matrices resembles the integral update equation (26). Although other valid choices for the value of p are possible, this is also the one that minimizes the norm of the wavelet coefficients.
- using the update weights (UW), i.e.

$$\omega_{j,\mathbf{v}_{j+1,e_i}}(\mathbf{v}_{j+1,o_l}) = \frac{u_{j,\mathbf{v}_{j+1,e_i}}(\mathbf{v}_{j+1,o_l})}{\sum_{\mathbf{v}_{j+1,e_k} \in \mathcal{N}_v^1(\mathbf{v}_{j+1,o_l})} u_{j,\mathbf{v}_{j+1,e_k}}(\mathbf{v}_{j+1,o_l})}, \quad (36)$$

  since these weights already describe a means for compensating detail loss in a manner resembling equation (17).

5. Discussion and numerical experiments

We propose examining the approximation quality of this lifting scheme design by computing the root mean square error (abbreviated as RMSE) of the even scaling coefficients. The evolution of this error is inversely proportional to the level of resolution if the notation used throughout this paper is maintained. However, the smaller the values, the better the filter design of the lifting scheme. Since there are two possible choices for the quadric error matrix redistribution weights (IDW and UW, respectively), the difference in approximation quality between these two strategies is further explored. In this respect, we have selected three LiDAR scans of various terrain regions. The first is a fragment of the Great Smoky Mountains (available through the `http://www.opentopography.org` portal) having a density of approximately 2.23 points per square meter and a total amount of 280,000 points (figure 1).

The remaining two sets are fragments of the Romanian Carpathian Mountains: one from the Fundata region (figure 2), having a density of 20 points per square meter, and a point count of 9 million, the other from the Iezer Mountains (figure 3), having a density of 5 points per square meter and a point count of 11.5 million). All sets contain, besides the geometric coordinates of the points, vegetation related attributes stored in separate scalar channels:

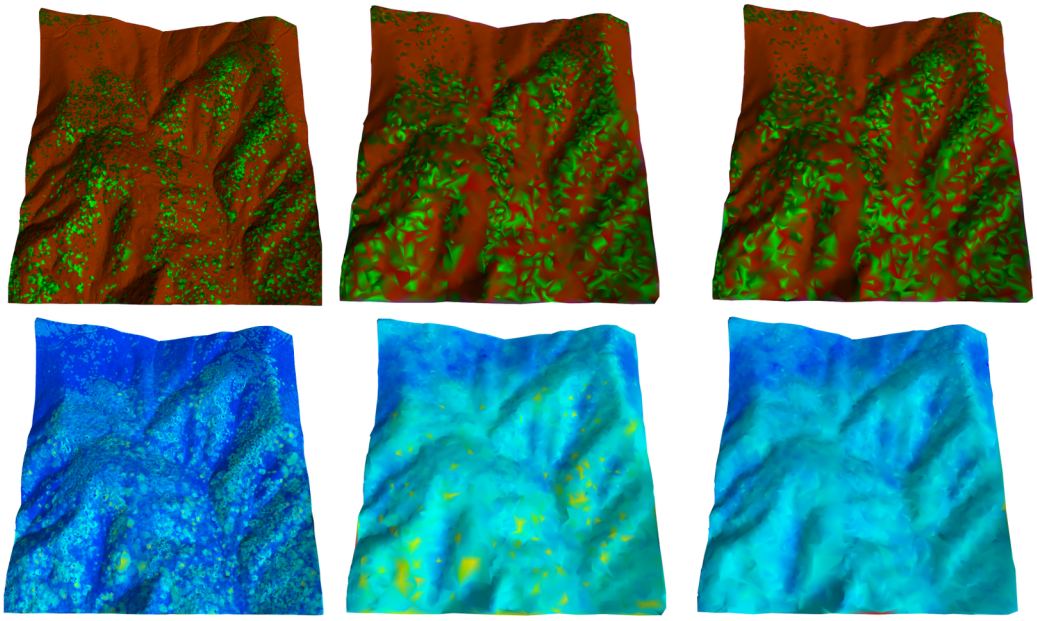

FIG. 1. Smoky data set subjected to a sequence of 12 analysis steps. The columns, in left to right order: the original mesh, lowest resolution using IDW and using UW, respectively. The bottom row is a temperature colormap generated using equation (12).

one for the vegetation class (an integer between 0 and 20) and one for the vegetation height (an integer between 0 and 255).

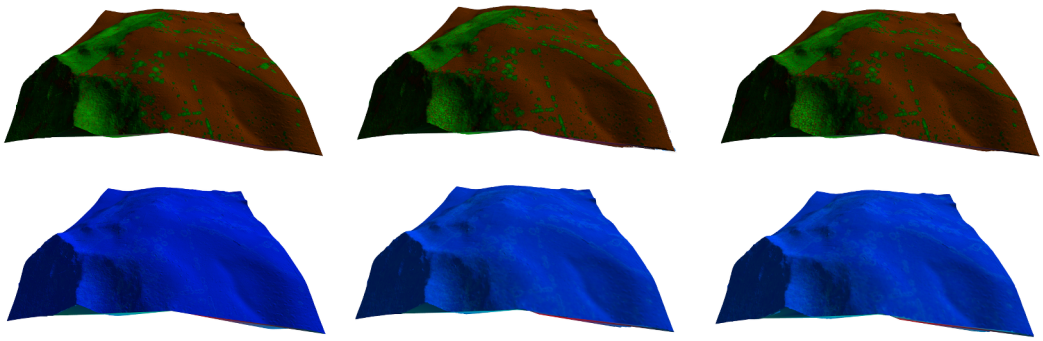

FIG. 2. Fundata data set subjected to a sequence of 12 analysis steps. The meaning of each row and column is given in figure 1.

The numerical experiments were carried out by downsampling the initial sets by repeatedly applying the analysis equations (31) and (33) twelve times.

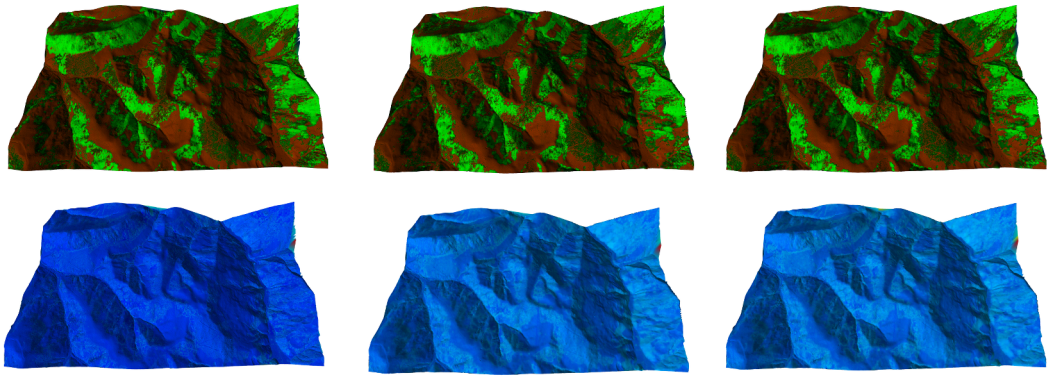

FIG. 3. Iezer data set subjected to a sequence of 12 analysis steps. The meaning of each row and column is given in figure 1.

TABLE 1. RMSE evolution during 12 successive analysis/simplification steps.

Level	Smoky		Fundata		Iezer	
	IDW	UW	IDW	UW	IDW	UW
12	0.0997648	0.0997648	0.0425317	0.0425317	0.0521839	0.0521839
11	0.246961	0.255929	0.103035	0.104011	0.131526	0.133092
10	0.463553	0.469643	0.185778	0.186109	0.249424	0.254507
9	0.749602	0.766483	0.297856	0.296403	0.430857	0.443759
8	1.15003	1.17577	0.463349	0.463635	0.721715	0.738768
7	1.84325	1.7946	0.730597	0.735497	1.15317	1.16666
6	2.68166	2.57539	1.15276	1.17955	1.70032	1.72003
5	3.71708	8 3.52738	1.81591	1.84962	2.34638	2.34925
4	4.9983	9 4.96675	2.721	2.78489	3.03661	3.03775
3	6.72188	6.70461	3.9207	3.95601	3.80415	3.82299
2	8.44014	8.4718	5.36374	5.44715	4.67153	4.69569
1	10.6596	11.0502	7.15283	7.27347	5.71604	5.77769

The RMSE value at level j was computed as

$$\mathrm{RMSE}_j = \frac{1}{|V_j|}\sqrt{\sum_{\mathbf{v}\in V_j} \|\mathbf{s}_{j,\mathbf{v}} - \mathbf{s}_{\mathrm{orig},\mathbf{v}}\|}, \tag{37}$$

where $j \in \overline{1,12}$ and $\mathbf{s}_{\mathrm{orig},\mathbf{v}}$ denotes the scaling vector of $\mathbf{v}$ at the highest level of resolution (i.e. the original mesh). The results of using both the IDW and UW weights are summed in table 1, where the RMSE_j values are presented for all three data sets.

6. Conclusions

In this work we explored the hypothesis of coupling the quadric error metric feature-preserving simplification mechanism with a second generation lifting scheme. This design, extended from the one-dimensional scenario, was used to downsample dense LiDAR meshes containing both geometry and vegetation attribute information. Instead of summing quadric error matrices, as required during the fusion of two vertices, the removal of odd vertex samples was followed by the redistribution of their corresponding matrices among the remaining even samples. Two redistribution weight designs were proposed, one based on the inverse distance (IDW) and the other on the update filter (UW). We have experimentally determined the inverse distance weights to produce better approximations and, consequently, to be the more suitable quadric error matrix redistribution mechanism.

Possible future work directions may involve smoothing and denoising applications as well as clustering and classification based on the wavelet coefficient distribution.

REFERENCES

[1] *W. J. Schroeder, J. A. Zarge and W. E. Lorensen*, Decimation of triangle meshes. SIGGRAPH Comput. Graph. **26**, 2 (1992), 65-70. DOI=10.1145/142920.134010 http://doi.acm.org/10.1145/142920.134010

[2] *W. Sweldens*, The lifting scheme: a construction of second generation wavelets. SIAM J. Math. Anal. **29**, 2 (1998), 511-546. DOI=10.1137/S0036141095289051 http://dx.doi.org/10.1137/S0036141095289051

[3] *T. Cioaca, B. Dumitrescu, M.-S. Stupariu*, Graph-based Wavelet Representation of Multi-variate Terrain Data, Comput. Graph. Forum. (in press)

[4] *T. Cioaca, B. Dumitrescu, M.-S. Stupariu, I. Patru-Stupariu, M. Naparus, I. Stoicescu, A. Peringer, A. Buttler, F. Golay*, Heuristic-driven Graph Wavelet Modeling of Complex Terrain, Proceedings of SPIE Vol. **9443**, 94431Y (2015)

[5] *M.l Garland and P. S. Heckbert*, Simplifying surfaces with color and texture using quadric error metrics. In Proc. of (VIS '98). IEEE Computer Society Press, Los Alamitos, CA, USA, 263-269.

[6] *M. Jansen*, Multiscale Local Polynomial Smoothing in a Lifted Pyramid for Non-Equispaced Data. Trans. Sig. Proc. **61**, 3 (2013), 545-555. DOI=10.1109/TSP.2012.2225059 http://dx.doi.org/10.1109/TSP.2012.2225059

[7] *S. Valette and R. Prost*, Wavelet-Based Progressive Compression Scheme for Triangle Meshes: Wavemesh. IEEE Transactions on Visualization and Computer Graphics **10**, 2 (2004), 123-129. DOI=10.1109/TVCG.2004.1260764 http://dx.doi.org/10.1109/TVCG.2004.1260764

[8] *M. Bertram*, Wavelet analysis for progressive meshes. In Proc. of the 23rd Spring Conf. on Comput. Graph. (2007), 161-167. DOI=10.1145/2614348.2614371 http://doi.acm.org/10.1145/2614348.2614371

[9] *Stefanoiu D., Stanasila O., Popescu D.*, Wavelets. Theory and Applications., Romanian Academy Press, Bucharest, Romania, 2010 (in Romanian)

[10] *C. Zhao and H. Sun*, Optimal Interpolatory Wavelets Transform for Multiresolution Triangular Meshes. Appl. Math., **2** 3 (2011), 369-378. DOI=10.4236/am.2011.23044.

[11] *D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, P. Vandergheynst*, The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains. IEEE Sig. Proc. Mag. **30**, 3 (2013), 83-98

[12] *K. Fujiwara*, Growth and the spectrum of the Laplacian of an infinite graph, Tohoku Math. J. (2) **48**, 2 (1996), 293-302. DOI=10.2748/tmj/1178225382

[13] *D. Shepard*, A two-dimensional interpolation function for irregularly-spaced data, Proc. 23rd National Conference ACM, ACM (1968) , 517-524.