Easy Path Wavelet Transform on Triangulations of the Sphere[1]

by

Gerlind Plonka[2] and Daniela Roşca[3]

[1] Received ; accepted

[2] Department of Mathematics, University of Duisburg-Essen, Campus Duisburg,
47048 Duisburg, Germany, email: gerlind.plonka@uni-due.de

[3] Department of Mathematics, Technical University of Cluj-Napoca, str. Daicoviciu 15,
400020 Cluj-Napoca, Romania, email: Daniela.Rosca@math.utcluj.ro

Corresponding Author:
Gerlind Plonka
Department of Mathematics, University of Duisburg-Essen
Campus Duisburg, 47048 Duisburg, Germany
Phone + 49 203 379 2677
fax + 49 203 379 2689
email gerlind.plonka@uni-due.de

Abstract

The computation of sparse representations of data on the sphere (as e.g. topographical data) is a crucial step for further processing such as multiple separation, migration, imaging and sparsity promoting data-recovery. The Easy Path Wavelet Transform (EPWT) is a new tool for sparse data representation that has recently been introduced for image compression. In this paper we consider the EPWT on spherical triangulations. It is a locally adaptive transform that works along pathways through the array of function values and exploits the local correlations of the data in a simple appropriate manner. In our approach the usual discrete one-dimensional orthogonal or biorthogonal wavelet transform can be applied. The EPWT can be used for defining a multiresolution analysis on the sphere in which the scaling spaces and the wavelet spaces depend adaptively on the data. Issues of implementation of the EPWT are also considered.

KEY WORDS. wavelet transform along pathways, data compression on the sphere, spherical triangulations.

1 Introduction

Today, we are faced with high amounts of data being generated by satellite observations that need to be adequately processed and analyzed. Therefore, one important problem in data analysis in geosciences is to construct efficient low-level representations using only a very small part of the original data. However, these sparse approximations should provide a precise characterization of relevant features of the data like discontinuities (edges) and texture components. In this paper, we propose a new transform for sparse representation of data on spherical surfaces, the EPWT. This new tool can be applied for better data analysis and data compression for problems in planetary science such as surface topography and analysis of geoid of other bodies.

It is well-known that wavelets can represent piecewise smooth signals efficiently. However, higher-dimensional structures may not be represented suitably by sparse wavelet decompositions based on tensor product wavelets, because directional geometrical properties of the data cannot be adapted. The last years have seen many attempts to construct locally adaptive wavelet-based schemes that take into account the special geometry of the data. In particular, for sparse representation of images, different ideas have been developed that try to exploit the local correlations of the data, see e.g. Claypoole et al (2003), Cohen and Matei (2001), Dekel and Leviatan (2006), Ding et al (2007), Donoho (1999), Mallat (2009), Plonka (2009), Shukla et al (2005).

We will especially focus on the EPWT recently introduced in Plonka (2009) for sparse image representation. In this paper, we want to adapt the EPWT to triangulations of the sphere. For this purpose, we apply the idea used in Roşca (2005a, 2005b) to obtain a suitable spherical triangulation. We employ a polyhedral subdivision domain. The triangular faces of the polyhedron are successively subdivided into four smaller triangles. Each triangle can be transported radially to the sphere. This approach has been used in Roşca (2005a, 2005b) for the construction of Haar wavelets and of locally supported rational spline wavelets on the sphere. Different wavelet constructions on triangulations of the sphere have been also studied by Schroeder and Sweldens (1995).

The idea of the EPWT on spherical triangulations is very simple. First we fix a certain neighborhood of a triangle, e.g. the three triangles that have common edges with the reference triangle. Next, we use a one-dimensional indexing of all triangles of the fixed triangulation and assume that each function value of a given data vector is associated to one triangle or rather its corresponding (one-dimensional) index. Now, in the first step we select a path through the complete index set in such a way that data points associated to neighboring indices in the path are strongly correlated. For that purpose, for each index we choose "the best" neighbor index that has not been used yet in the path, such that the absolute difference between neighboring data values is the smallest. The complete path vector can be seen as a permutation of the original index vector. Then we apply a suitable (one-dimensional) discrete wavelet transform to the data vector along the path, and the choice of the path will ensure that most wavelet coefficients remain small. The same procedure can be successively applied to the down-sampled data. After a suitable number of iterations, we apply a shrinkage procedure to all wavelet coefficients in order to find a sparse digital representation of the function. For reconstruction one needs the path vector at each level in order to apply the inverse wavelet transform.

The paper is organized as follows. In Section 2 we shall give a detailed description of the EPWT on triangulations of the sphere. In Section 3 we present an interpretation of the algorithm for adaptive Haar wavelets on spherical triangulations based on a multiresolution structure. Finally, Section 4 is devoted to implementation issues and numerical experiments. We shall give an explicit description of a bijective mapping of a given triangulation to a one-dimensional index set, and we shall explicitly determine a neighborhood of a given index in this one-dimensional index set.

2 EPWT on triangulations of the sphere

The main idea of the new algorithm is to exploit the local correlations of the data in an efficient way.

2.1 Spatial and spherical triangulations

We consider the unit sphere $\mathbb{S}^2$ of $\mathbb{R}^3$ with the center O. Let Π be a convex polyhedron with O inside and with triangular faces. For example we can take an icosahedron, a cube with triangulated faces, an octahedron, etc. The surface (boundary) of the polyhedron will be denoted by Ω. We denote by $\mathcal{T} = \mathcal{T}^0 = \{T_1, \ldots, T_M\}$ the set of faces of Π and by V^0 the set of vertices. By recursive subdivision we obtain a uniform refinement of $\mathcal{T}^0$. For each triangle $[M_1M_2M_3]$ with vertices M_1, M_2, M_3 in V^0, let A_1, A_2, A_3 denote the midpoints of the edges M_2M_3, M_3M_1 and M_1M_2, respectively. Then we consider the refined triangulation of $\mathcal{T}^0$

$$\mathcal{T}^1 = \bigcup_{[M_1M_2M_3]\in\mathcal{T}^0} \{[M_1A_2A_3], [A_1M_2A_3], [A_1A_2M_3], [A_1A_2A_3]\}.$$

Continuing the refinement process in the same way, we obtain a triangulation $\mathcal{T}^j$ of Ω for $j \in \mathbb{N}$. For application of the EPWT we will stop the refinement process at a suitable sufficiently high (fixed) level j depending on the data set in the application. We define the set of vertices V^j accordingly. In particular, each triangle in $\mathcal{T}^j$ is uniquely determined by its three vertices. For application of the EPWT we will also need a one-dimensional index set $J = J^j$ for the triangles in $\mathcal{T}^j$. Using the octahedron, for $j = 1$, this one-dimensional index set J can be assigned as shown in Figure 1 (right). Observe that for the octahedron the number of triangles at the jth level is given by $\#J = \#\mathcal{T}^j = 2^{2j+3}$.

Let $\langle\cdot,\cdot\rangle_\Omega$ be the following inner product based on the given polyhedron

$$\langle f,\, g\rangle_\Omega := \sum_{T\in\mathcal{T}^0} \frac{1}{a(T)} \int_T f(x)\, g(x)\, dx, \qquad f, g \in L^2(\Omega),$$

where $a(T)$ is the area of the triangle T. As shown in Roşca (2005b), the induced norm $\|\cdot\|_\Omega = \langle\cdot,\cdot\rangle_\Omega^{1/2}$ is equivalent to the usual norm in $L^2(\Omega)$. In order to obtain a spherical triangulation, we now define for the given polyhedron Π the radial projection onto $\mathbb{S}^2$, $p : \Omega \to \mathbb{S}^2$,

$$p(x, y, z) = \left(\frac{x}{\sqrt{x^2+y^2+z^2}}, \frac{y}{\sqrt{x^2+y^2+z^2}}, \frac{z}{\sqrt{x^2+y^2+z^2}}\right), \quad (x, y, z) \in \Omega.$$

The inverse mapping $p^{-1} : \mathbb{S}^2 \to \Omega$ is given by

$$p^{-1}(\eta_1, \eta_2, \eta_3) = \left(-\frac{\eta_1 d}{a\eta_1 + b\eta_2 + c\eta_3}, -\frac{\eta_2 d}{a\eta_1 + b\eta_2 + c\eta_3}, -\frac{\eta_3 d}{a\eta_1 + b\eta_2 + c\eta_3}\right),$$

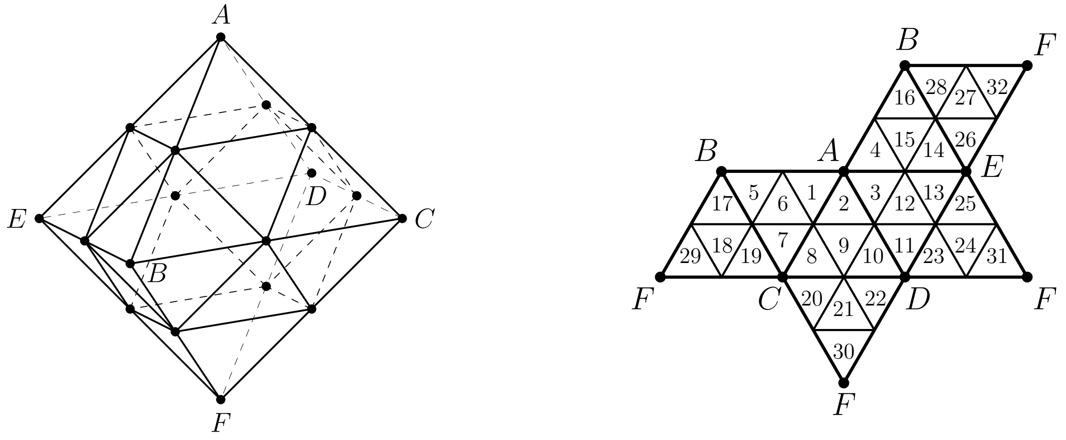

Figure 1. Illustration of the octahedron with $\mathcal{T}^1$ triangulation (left) and a fold apart version of the octahedron on the plane with a one-dimensional indexing of all faces.

where $ax+by+cz+d=0$ is the equation of the face of Π onto which the point $(\eta_1, \eta_2, \eta_3) \in \mathbb{S}^2$ projects. If the point $(\eta_1, \eta_2, \eta_3) \in \mathbb{S}^2$ projects onto an edge, then we may choose one of the adjacent faces to express the function p^{-1}. Considering the images $U_i = p(T_i)$ of the triangles $T_i \in \mathcal{T}^j$ under the projection p, we say that $\mathcal{U}^j = \{U_i = p(T_i),\, T_i \in \mathcal{T}^j\}$ is a triangulation of the sphere $\mathbb{S}^2$.

For L^2-integrable functions on the sphere $\mathbb{S}^2$, let the scalar product $\langle \cdot, \cdot \rangle_*$ be defined by

$$\langle F, G \rangle_* := \langle F \circ p, G \circ p \rangle_\Omega.$$

Again, one can show that the corresponding norm $\| \cdot \|_* = \langle \cdot, \cdot \rangle_*^{1/2}$ is equivalent to the usual norm $\| \cdot \|_2$ of $L^2(\mathbb{S}^2)$, see Roşca (2005b, 2009).

We shall apply the EPWT especially to piecewise constant functions on spherical triangulations. For indexing the spherical triangles in $\mathcal{U}^j$, we use the same index set $J = J^j$ as for the triangulation $\mathcal{T}^j$ of the polyhedron. Now, for a fixed spherical triangle $U_i \in \mathcal{U}^j$, $i \in J$, we define the piecewise constant function $\chi_{U_i} : \mathbb{S}^2 \to \mathbb{R}$,

$$\chi_{U_i} = \begin{cases} 1, & \text{inside the triangle}\, U_i, \\ 0, & \text{elsewhere.} \end{cases}$$

Then we can consider the space $\mathcal{V} = \text{span}\,\{\chi_{U_i},\ i \in J\}$ that consists of the piecewise constant functions on the triangles of $\mathcal{U}^j$.

2.2 Definitions and Notations for the EPWT

In order to explain the idea of the EPWT, where we want to use the discrete one-dimensional wavelet transform along *path vectors* through the data, we first need some further definitions and notations.

Let us assume that we have given a fixed refined spherical triangulation $\mathcal{U}^j$ with a corresponding set of vertices $p(V^j)$. Further, let J be a one-dimensional index set for the spherical triangles in $\mathcal{U}^j$. We determine a *neighborhood* of an index $\nu \in J$ of the set of spherical triangles by

$$\mathcal{N}(\nu) = \{\mu \in J \setminus \{\nu\} : T_\mu \text{ and } T_\nu \text{ have a common edge}\}.$$

Hence each index $\nu \in J$ has exactly three neighbors. One may also use a bigger neighborhood, e.g. $\mathcal{N}(\nu) = \{\mu \in J \setminus \{\nu\} : T_\mu$ and T_ν have a common edge or a common vertex$\}$, in this case each index has 12 neighbors. We also need the definition of neighborhood of subsets of an index set. We shall consider disjoint *partitions* of J of the form $\{J_1, J_2, \ldots, J_r\}$, where $J_\mu \cap J_\nu = \emptyset$ for $\mu \neq \nu$ and $\bigcup_{\nu=1}^r J_\nu = J$. We then say that two different subsets J_ν and J_μ from the partition are *neighbors*, and we write $J_\nu \in \mathcal{N}(J_\mu)$, if there exist an index $l \in J_\nu$ and an index $l_1 \in J_\mu$ such that $l \in \mathcal{N}(l_1)$.

We consider a piecewise constant function $f \in \mathcal{V}$, i.e., we identify each spherical triangle in $\mathcal{U}^j$ with a function value of f. Hence f is uniquely determined by the data vector $(f_\nu)_{\nu \in J}$. We will look for path vectors through index subsets of J and apply a one-dimensional wavelet transform along these path vectors. Any orthogonal or biorthogonal one-dimensional wavelet transform can be used here.

We say that a vector of indices $(l_k, \ldots, l_{k+n})$, $1 \leq k < k+n \leq \#J$, is *connected*, if we have $l_{\nu+1} \in \mathcal{N}(l_\nu)$ for $\nu = k, \ldots, k+n-1$. Here, $\#J$ denotes the cardinality of J. Such a connected index vector is called *pathway.* We are interested in a *complete path* through the index set J. A complete path $\mathbf{p}$ through J is a vector containing all indices of J in a certain order, i.e. $\mathbf{p} \in \mathbb{Z}^{\#J}$ is a permutation of $(1, 2, \ldots, \#J)$. This complete path should be composed by a number of pathways, i.e. $\mathbf{p} = (\mathbf{p}_1; \ldots; \mathbf{p}_r)$, where $\mathbf{p}_\nu$, $\nu = 1, \ldots, r$, are connected vectors of indices.

One simple example of such a complete path is to take just $\mathbf{p} = (1, 2, \ldots, \#J)$. Using the indexing as in Figure 1, this path has interruptions after indices 4, 16 and 28. It is also possible to take a path without interruptions here, e.g. $(1, 2, 3, 4, 15, 16, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 28, 27, 32, 31, 30, 29)$.

2.3 Description of the EPWT

In this subsection we rely on the detailed description of the EPWT in Plonka (2009), and give a summary of the idea. We start with the decomposition of the real data $(f_\nu)_{\nu \in J}$, and we assume that $N = \#J$ is a multiple of 2^L with $L \in \mathbb{N}$. For the considered octahedron we have $N = 2^{2j+3}$. Then we will be able to apply L levels of the EPWT.

Decomposition

First level

We first determine a complete path vector $\mathbf{p}^L$ through the index set $J = \{1, 2, \ldots, N\}$ and then apply a suitable discrete one-dimensional (periodic) wavelet transform to the function values along this path $\mathbf{p}^L$.

We start with $\mathbf{p}^L(1) = 1$. In order to determine the second index $\mathbf{p}^L(2)$, we seek the minimum of absolute differences of the function values corresponding to the neighborhood of the index 1, and put

$$\mathbf{p}^L(2) = \underset{k}{\operatorname{argmin}} \{|\mathbf{f}^L(1) - \mathbf{f}^L(k)|, k \in \mathcal{N}(1)\}.$$

We proceed in this manner, thereby determining a path vector through the index set J that is locally adapted to the function f (easy path). With the procedure above we obtain a pathway such that the absolute differences between neighboring function values $\mathbf{f}^L(l)$ along the path stay as small as possible. Generally, having given the index $\mathbf{p}^L(l)$, $1 \leq l \leq N-1$, we determine the next value $\mathbf{p}^L(l+1)$ by

$$\mathbf{p}^L(l+1) = \operatorname*{argmin}_k \{|\mathbf{f}^L(\mathbf{p}^L(l)) - \mathbf{f}^L(k)|, k \in \mathcal{N}(\mathbf{p}^L(l)), k \neq \mathbf{p}^L(\nu),\ \nu = 1, \ldots, l\}.$$

It can happen that the choice of the next index value $\mathbf{p}^L(l+1)$ is not unique, if the above minimum is attained by more than one index. In this case, one may just fix favorite directions in order to determine a unique pathway.

Another situation can occur during the procedure, namely that all indices in the neighborhood of a considered index $\mathbf{p}^L(l)$ have already been used in the path $\mathbf{p}^L$. In this case we need to start with a new pathway, i.e., we have an interruption in the path vector. We need to choose one index $\mathbf{p}^L(l+1)$ from the remaining indices in J that have not yet been taken in $\mathbf{p}^L$. There are different possibilities for starting the next pathway. One simple choice is to take just the smallest index from J that has not been used so far. Another choice is to look for a next index, such that again the absolute difference $|\mathbf{f}^L(\mathbf{p}^L(l)) - \mathbf{f}^L(\mathbf{p}^L(l+1))|$ is minimal, i.e., we take in this case

$$\mathbf{p}^L(l+1) = \operatorname*{argmin}_k \{|\mathbf{f}^L(\mathbf{p}^L(l)) - \mathbf{f}^L(k)|, k \in J, k \neq \mathbf{p}^L(\nu),\ \nu = 1, \ldots, l\}.$$

We proceed in this manner and obtain finally a complete path vector $\mathbf{p}^L \in \mathbb{Z}^N$ that is a permutation of $(1, 2, \ldots, N)$.

After having constructed the path $\mathbf{p}^L$, we apply one level of the discrete one-dimensional Haar wavelet transform or any other discrete orthogonal or biorthogonal periodic wavelet transform to the vector of function values $(\mathbf{f}^L(\mathbf{p}^L(l)))_{l=1}^N$ along the path $\mathbf{p}^L$. We find the vector $\mathbf{f}^{L-1} \in \mathbb{R}^{N/2}$ containing the low-pass part and the vector of wavelet coefficients $\mathbf{g}^{L-1} \in \mathbb{R}^{N/2}$. While the wavelet coefficients will be stored in $\mathbf{g}^{L-1}$, we proceed further with the low-pass vector $\mathbf{f}^{L-1}$ at the second level.

Remark. For the numerical implementation, the method for choosing the start index of a new pathway has essential consequences. The simple method to take the smallest "free" index is also the cheapest with regard to the storing costs for the path, but for each new pathway one may produce a big wavelet coefficient. The second method (preferred here), avoids big wavelet coefficients but leads to higher costs for storing the path $\mathbf{p}^L$. Plonka (2009) has proposed an efficient method for index selections, that finds a compromise between these issues.

Second level

Let us consider the index sets $J_l^{L-1} := \{\mathbf{p}^L(2l-1), \mathbf{p}^L(2l)\}$, $l = 1, \ldots, N/2$, that determine a partition of J. Now, each such index set J_l^{L-1} corresponds to the function value $\mathbf{f}^{L-1}(l)$. We repeat the same procedure as in the first step, but replacing the single indices with corresponding function values by the new index sets J_l^{L-1} and the corresponding smoothed function values $\mathbf{f}^{L-1}(l)$. The new path vector $\mathbf{p}^{L-1} \in \mathbb{Z}^{N/2}$ should now be a permutation of $(1, 2, \ldots, N/2)$. We start again with the first index set J_1^{L-1}, i.e., $\mathbf{p}^{L-1}(1) = 1$. Having already found $\mathbf{p}^{L-1}(l)$, $1 \leq l \leq N/2 - 1$, we determine the next value $\mathbf{p}^{L-1}(l+1)$ by

$$\begin{aligned}\mathbf{p}^{L-1}(l+1) \quad &= \quad \underset{k}{\operatorname{argmin}} \, \{|\mathbf{f}^{L-1}(\mathbf{p}^{L-1}(l)) - \mathbf{f}^{L-1}(k)|, \, J_k^{L-1} \in \mathcal{N}(J_{\mathbf{p}^{L-1}(l)}^{L-1}), \\ & \qquad\qquad k \neq \mathbf{p}^{L-1}(\nu), \, \nu = 1, \ldots, l\}.\end{aligned}$$

If the new value $\mathbf{p}^{L-1}(l+1)$ is not uniquely determined by the minimizing procedure, we can just fix favorite directions in order to obtain a unique path. If for the set $J_{\mathbf{p}^{L-1}(l)}^{L-1}$ there is no neighboring index set that has not been used yet in the path vector $\mathbf{p}^{L-1}$, then we have to interrupt the path and to find a new good index set (that has been not used so far) to start a new pathway. As in the first level, we try to keep the differences of function values along the path small and may choose in this case

$$\begin{aligned}\mathbf{p}^{L-1}(l+1) \quad &= \quad \underset{k}{\operatorname{argmin}} \, \{|\mathbf{f}^{L-1}(\mathbf{p}^{L-1}(l)) - \mathbf{f}^{L-1}(k)|, 1 \leq k \leq N/2, \\ & \qquad\qquad k \neq \mathbf{p}^{L-1}(\nu), \; \nu = 1, \ldots, l\}.\end{aligned}$$

After having completed the path vector $\mathbf{p}^{L-1}$, we apply again the chosen discrete (periodic) wavelet transform to the vector $(\mathbf{f}^{L-1}(\mathbf{p}^{L-1}(l)))_{l=1}^{N/2}$ along the path $\mathbf{p}^{L-1}$. Assuming that $N/4 \in \mathbb{N}$, i.e. $L \geq 2$, we obtain the vector $\mathbf{f}^{L-2} \in \mathbb{R}^{N/4}$ containing the low-pass part and the vector of wavelet coefficients $\mathbf{g}^{L-2} \in \mathbb{R}^{N/4}$. While the wavelet coefficients in $\mathbf{g}^{L-2}$ will be stored, we proceed now again with the low-pass vector $\mathbf{f}^{L-2}$ in the next step.

Further levels

If N is of the form $2^L s$ with $s \in \mathbb{N}$ being greater than or equal to the lengths of low-pass and high-pass filters in the chosen discrete wavelet transform, then we may apply the procedure L times. For a given vector $\mathbf{f}^{L-j}$, $0 < j < L$, we consider in the $(j+1)$-th step the index sets

$$J_l^{L-j} = J_{\mathbf{p}^{L-j+1}(2l-1)}^{L-j+1} \cup J_{\mathbf{p}^{L-j+1}(2l)}^{L-j+1}, \qquad l = 1, \ldots, N/2^j,$$

with the corresponding function values $\mathbf{f}^{L-j}(l)$. Then we determine a path vector $\mathbf{p}^{L-j}$ of length $N/2^j$ as a permutation of $(1, 2, \ldots, N/2^j)$, as before. Finally, we apply the (periodic) wavelet transform to the vector $(\mathbf{f}^{L-j}(\mathbf{p}^{L-j}(l)))_{l=1}^{N/2^j}$ along the path $\mathbf{p}^{L-j}$, thereby obtaining the low-pass vector $\mathbf{f}^{L-j-1} \in \mathbb{R}^{N/2^{j+1}}$ and the vector of wavelet coefficients $\mathbf{g}^{L-j-1} \in \mathbb{R}^{N/2^{j+1}}$.

Output

As output of the complete procedure after L iterations we obtain the coefficient vector

$$\mathbf{g} = (\mathbf{f}^0, \mathbf{g}^0, \mathbf{g}^1, \ldots, \mathbf{g}^{L-1}) \in \mathbb{R}^N$$

and the vector determining the paths in each iteration step

$$\mathbf{p} = (\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^L) \in \mathbb{R}^{2N(1-1/2^L)}.$$

These two vectors contain the entire information about the original function f.

In order to find a sparse representation of f, we apply a *shrinkage procedure* to the wavelet coefficients in the vectors $\mathbf{g}^j$, $j = 0, \ldots, L-1$, and obtain $\widetilde{\mathbf{g}}^j$. Here one can use the hard threshold function

$$S_\sigma(x) = \begin{cases} x, & |x| \geq \sigma, \\ 0, & |x| < \sigma. \end{cases}$$

Reconstruction

The reconstruction of (the approximation) $\tilde{f}$ resp. $\widetilde{\mathbf{f}}^L$ from $\tilde{\mathbf{g}}$ and $\mathbf{p}$ is given as follows.

```
For $j = 0$ to $L - 1$ do
    Apply the inverse discrete wavelet transform to the vector $(\widetilde{\mathbf{f}}^j, \widetilde{\mathbf{g}}^j) \in \mathbb{R}^{r2^j}$
    (with $\widetilde{\mathbf{f}}^0 := \mathbf{f}^0$) in order to obtain $\widetilde{\mathbf{f}}_p^{j+1} \in \mathbb{R}^{r2^{j+1}}$.
    Apply the permutation
        $\widetilde{\mathbf{f}}^{j+1}(\mathbf{p}^{j+1}(k)) = \widetilde{\mathbf{f}}_p^{j+1}(k), \qquad k = 1, \ldots, r2^{j+1}.$
end.
```

2.4 Example

We illustrate the idea of function decomposition with the EPWT on the sphere in the following small example. Let a set of 32 function values be given on the sphere, where each function value corresponds to a spherical triangle on the sphere that has been obtained by radial projection of the triangulated octahedron in Figure 1 (left). The values are given as a vector of length 32 corresponding to the one-dimensional indexing of the triangles in Figure 1 (right),

$$\begin{aligned} \mathbf{f} \;=\; & (0.4492,\, 0.4219,\, 0.4258,\, 0.4375,\, 0.4141,\, 0.4531,\, 0.4180,\, 0.4258, \\ & 0.4375,\, 0.4292,\, 0.4219,\, 0.4219,\, 0.4219,\, 0.4258,\, 0.4023,\, 0.4141, \\ & 0.4219,\, 0.4219,\, 0.4297,\, 0.4375,\, 0.4141,\, 0.4023,\, 0.4258,\, 0.4219, \\ & 0.4258,\, 0.4180,\, 0.4531,\, 0.4141,\, 0.4375,\, 0.4258,\, 0.4219,\, 0.4492). \end{aligned}$$

Starting with the index 1, we now determine the first path vector. Index 1 with the function value 0.4492 has the neighbors $2, 4$ and 6 with corresponding values 0.4219, 0.4375 and 0.4531, respectively (see Figure 2). Hence, the second index in the path is 6. Proceeding further according to Section 2.3 we obtain

$$\begin{aligned} \mathbf{p}^5 \;=\; & (1, 6, 7, 8, 9, 10, 11, 12, 13, 14, 26, 25, 24, 31, 30, 21, 22, 23;\; 3, 2;\; 17, 18, 19, 20; \\ & 4, 15, 16, 5;\; 28, 27, 32, 29), \end{aligned}$$

where the pathways are separated by semicolons. This path has four interruptions and is illustrated by arrows in Figure 2 (left).

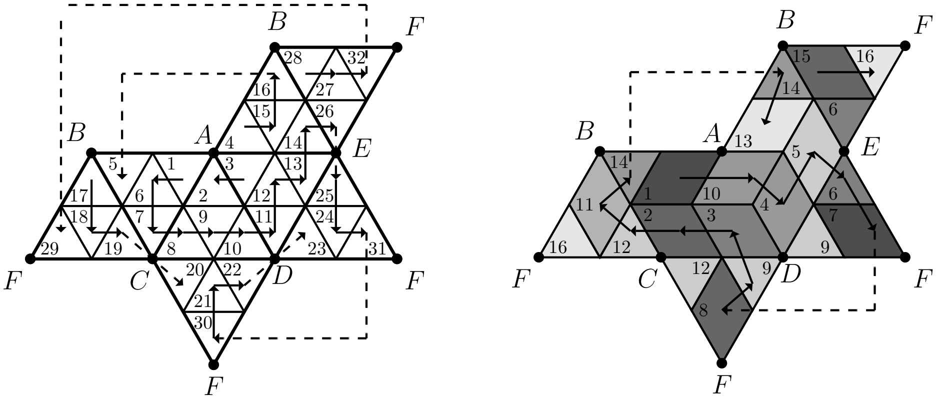

Figure 2. Illustration of first path through the $\mathcal{T}^1$ triangulation of the octahedron (left) and of the low-pass part after the first level of EPWT with Haar wavelet transform (right). Index sets at the second level are illustrated by different gray values, and path vectors are represented by arrows.

An application of the Haar wavelet transform (with unnormalized filter coefficients $h_0 = h_1 = 1/2$, $g_0 = 1/2$, $g_1 = -1/2$) along this path gives (with truncation after four digits) the low-pass coefficients

$$\begin{aligned}\mathbf{f}^4 \;&=\; (0.4512, 0.4219, 0.4334, 0.4219, 0.4238, 0.4219, 0.4219, 0.4200,\\ &\quad 0.4140, 0.4238, 0.4219, 0.4336, 0.4199, 0.4141, 0.4336, 0.4434),\end{aligned}$$

and the wavelet coefficients

$$\begin{aligned}\mathbf{g}^4 \;&=\; (-0.0020, -0.0039, -0.0042, 0., -0.0020, -0.0039, 0., 0.0058,\\ &\quad -0.0118, 0.0020, 0., -0.0039, 0.0176, 0., -0.0195, 0.0058).\end{aligned}$$

We now proceed to the second level. For the smoothed vector of function values $\mathbf{f}^4$ corresponding to the 16 index sets that are illustrated by gray values in Figure 2 (right), we obtain the next path

$$\mathbf{p}^4 = (1, 10, 4, 5, 6, 7, 8, 9, 3, 2, 12, 11, 14, 13;\ 15, 16),$$

illustrated by arrows in Figure 2 (right). The path $\mathbf{p}^4$ has one interruption. An application of the Haar wavelet transform along $\mathbf{p}^4$ gives

$$\begin{aligned}\mathbf{f}^3 \;&=\; (0.4375, 0.4229, 0.4219, 0.4170, 0.4276, 0.4278, 0.4170, 0.4385),\\ \mathbf{g}^3 \;&=\; (0.0136, -0.0010, 0., 0.0030, 0.0057, 0.0058, 0.0029, -0.0049).\end{aligned}$$

At the third level we start with the smoothed vector $\mathbf{f}^3$ corresponding to the 8 index sets that are illustrated by gray values in Figure 3 (left). We find now the path

$$\mathbf{p}^3 = (1, 5, 6, 8, 3, 2, 4;\ 7),$$

see Figure 3 (left). This leads to

$$\begin{aligned}\mathbf{f}^2 &= (0.4326, 0.4331, 0.4224, 0.4170),\\ \mathbf{g}^2 &= (0.0049, -0.0054, 0.0005, 0.).\end{aligned}$$

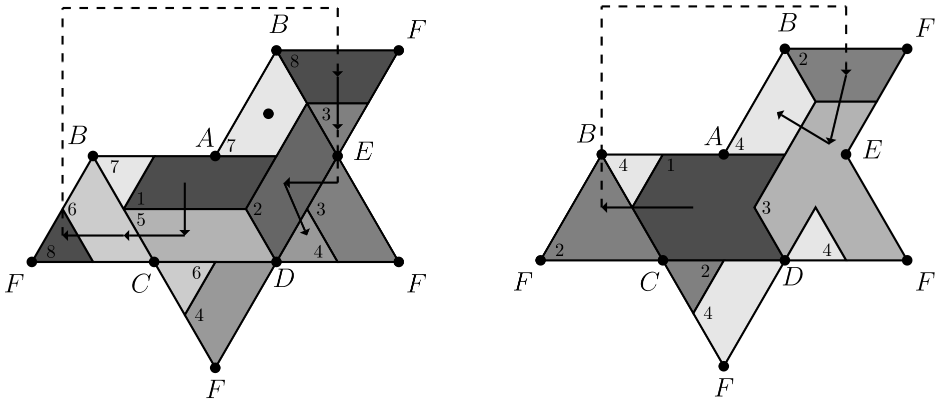

Figure 3. Illustration of the third path through the index sets of the octahedron (left) and of the fourth path of the obtained index sets (right) for EPWT with Haar wavelet transform. Index sets are illustrated by different gray values, and path vectors are represented by arrows.

In the fourth level we have only 4 index sets that correspond to the values in $\mathbf{f}^2$, see Figure 3 (right). Hence we find $\mathbf{p}^2 = (1, 2, 3, 4)$ and

$$\mathbf{f}^1 = (0.4328, 0.4197), \qquad \mathbf{g}^1 = (-0.0003,\ 0.0027).$$

Finally, with $\mathbf{p}^1 = (1, 2)$ the last transform yields $\mathbf{f}^0 = (0.4263)$ and $\mathbf{g}^0 = (0.0066)$.

3 Adaptive Haar wavelet bases on the sphere

We want to study the question, how the EPWT introduced in Section 2 can be understood in terms of a multiresolution analysis on the sphere. Let $N_j := \#J = 2^L\, s$ with $L,\, s \in \mathbb{N}$ be the number of triangles in the spherical triangulation $\mathcal{U}^j$ of $\mathbb{S}^2$. For the octahedron we can take $L = 2j + 3$ and $s = 1$. We consider the space

$$\mathcal{V} = \mathcal{V}^L = \operatorname{span}\{\phi_i^L = \chi_{U_i},\, U_i \in \mathcal{U}^j\} = \operatorname{span}\{\phi_i^L,\, i \in J\}$$

of piecewise constant functions on the fixed spherical triangulation $\mathcal{U}^j$. Obviously, $\{\phi_i^L : i \in J\}$ forms an orthonormal basis of $\mathcal{V}^L$ with respect to the scalar product $\langle\cdot,\cdot\rangle_*$ on $L^2(\mathbb{S}^2)$, defined in Section 2.1. Further, let $f \in \mathcal{V}^L$ be a given function of the form

$$f = \sum_{\mu \in J} a_\mu^L\, \phi_\mu^L.$$

Let us also fix the Haar wavelet filter bank given by the analysis filters h, g with $h(0) = h(1) = 1/\sqrt{2}$ and $g(0) = -1/\sqrt{2}$, $g(1) = 1/\sqrt{2}$. The synthesis filters for the orthonormal Haar wavelet filter bank are given by $\tilde{h}(0) = \tilde{h}(1) = 1/\sqrt{2}$ and $\tilde{g}(0) = -1/\sqrt{2}$, $\tilde{g}(1) = 1/\sqrt{2}$.

As we have seen in Section 2.3, applying the first level of the EPWT with Haar filters, we first need to determine a path $\mathbf{p}^L$, i.e., a permutation of $(1, \ldots, N_j)$, and then apply the one-dimensional Haar wavelet transform to the vector $(f(\mathbf{p}^L(\mu)))_{\mu=1}^{N_j}$. This transform can be interpreted by determining the new function spaces

$$\mathcal{V}^{L-1} = \text{span}\left\{\phi_i^{L-1} := \frac{1}{\sqrt{2}}\Big(\phi^L_{\mathbf{p}^L(2i-1)} + \phi^L_{\mathbf{p}^L(2i)}\Big),\ i = 1, \ldots, N_j/2\right\}$$

and

$$\mathcal{W}^{L-1} = \text{span}\left\{\psi_i^{L-1} := \frac{1}{\sqrt{2}}\Big(\phi^L_{\mathbf{p}^L(2i-1)} - \phi^L_{\mathbf{p}^L(2i)}\Big),\ i = 1, \ldots, N_j/2\right\}.$$

Obviously, $\mathcal{V}^{L-1}$ is generated by characteristic functions whose supports consist of two (usually) neighboring spherical triangles along the path $\mathbf{p}^L$. We find $\mathcal{V}^{L-1} \subset \mathcal{V}^L$ and $\mathcal{W}^{L-1} \subset \mathcal{V}^L$, and moreover, we have

$$\mathcal{V}^{L-1} + \mathcal{W}^{L-1} = \mathcal{V}^L \quad \text{and} \quad \mathcal{V}^{L-1} \perp \mathcal{W}^{L-1},$$

where the orthogonality holds with respect to the scalar product $\langle \cdot\,, \cdot \rangle_*$.

We can apply the EPWT transform at L levels, thereby adaptively defining scaling spaces and wavelet spaces according to the path vectors $\mathbf{p}^l$, $l = L, L-1, \ldots, 1$ of lengths $2^l s$. We obtain as before for $l = L, L-1, \ldots, 1$,

$$\mathcal{V}^{l-1} = \text{span}\left\{\phi_i^{l-1} := \frac{1}{\sqrt{2}}\Big(\phi^l_{\mathbf{p}^l(2i-1)} + \phi^l_{\mathbf{p}^l(2i)}\Big),\ i = 1, \ldots, 2^{l-1}s\right\},$$

and

$$\mathcal{W}^{l-1} = \text{span}\left\{\psi_i^{l-1} := \frac{1}{\sqrt{2}}\Big(\phi^l_{\mathbf{p}^l(2i-1)} - \phi^l_{\mathbf{p}^l(2i)}\Big),\ i = 1, \ldots, 2^{l-1}s\right\},$$

where again $\mathcal{V}^{l-1} + \mathcal{W}^{l-1} = \mathcal{V}^l$ and $\mathcal{V}^{l-1} \perp \mathcal{W}^{l-1}$ with respect to $\langle \cdot, \cdot \rangle_*$. Observe that the spaces $\mathcal{V}^l$ and $\mathcal{W}^l$ are adaptively dependent on the considered original function $f \in \mathcal{V}^L$.

With the Haar filters given above, we find the decomposition of $f = f^L \in \mathcal{V}^L$,

$$f^L = f^0 + \sum_{l=0}^{L-1} g^l,$$

with $f^0 \in \mathcal{V}^0$ and $g^l \in \mathcal{W}^l$ for $l = 0, \ldots, L-1$. More precisely, with the notation $f^{l+1} \in \mathcal{V}^{l+1}$ satisfying $f^{l+1} = f^l + g^l$ and with

$$f^{l+1} = \sum_{\mu=1}^{2^{l+1}r} a_\mu^{l+1}\,\phi_\mu^{l+1}, \qquad f^l = \sum_{\mu=1}^{2^l r} a_\mu^l\,\phi_\mu^l, \qquad g^l = \sum_{\mu=1}^{2^l r} b_\mu^l\,\psi_\mu^l,$$

the decomposition algorithm reads

$$a^l_\mu = \frac{1}{\sqrt{2}}\Big(a^{l+1}_{\mathbf{p}^{l+1}(2\mu-1)} + a^{l+1}_{\mathbf{p}^{l+1}(2\mu)}\Big), \quad b^l_\mu = \frac{1}{\sqrt{2}}\Big(a^{l+1}_{\mathbf{p}^{l+1}(2\mu-1)} - a^{l+1}_{\mathbf{p}^{l+1}(2\mu)}\Big), \quad \mu = 1, \ldots, 2^l s.$$

Vice versa, for $l = 0, \ldots, L-1$ the reconstruction algorithm is given by

$$a^{l+1}_{\mathbf{p}^{l+1}(2\mu-1)} = \frac{1}{\sqrt{2}}\Big(a^l_\mu + b^l_\mu\Big), \quad a^{l+1}_{\mathbf{p}^{l+1}(2\mu)} = \frac{1}{\sqrt{2}}\Big(a^l_\mu - b^l_\mu\Big), \quad \mu = 1, \ldots, 2^l s.$$

4 Implementation issues

In order to implement the EPWT for real data sets, one is confronted with the following issues.

1. Projection of the data set or the function on the spherical triangulation with a certain fixed resolution 2^{-j}, such that each triangle in $\mathcal{T}^j$ (resp. $\mathcal{U}^j$) corresponds to one data value.

2. Definition of a bijective mapping of the triangles in $\mathcal{T}^j$ (resp. $\mathcal{U}^j$) to a one-dimensional index set.

3. Determination of neighbors in the one-dimensional index set.

The first issue is not related to the EPWT but occurs for all applications of spherical triangulations (see e.g. Roşca (2005b)). We want to study here the second and the third issue.

4.1 Bijective mapping to a one-dimensional index set

Usually, we consider the subdivision process separately on each of the eight triangular faces of the octahedron. Let these faces be numbered by $d \in \{1, 2, \ldots, 8\}$, where the first four triangles are consecutively ordered on the northern part of the octahedron (i.e., all points of these triangles have a z-coordinate greater than or equal to zero), and the other 4 triangular faces, on the southern part of the octahedron, have numbers 5,6,7,8. Using Figure 1, let $T_1 = [B, C, A]$, $T_2 = [C, D, A]$, $T_3 = [D, E, A]$, $T_4 = [E, B, A]$, as well as $T_5 = [B, C, F]$, $T_6 = [C, D, F]$, $T_7 = [D, E, F]$, $T_8 = [E, B, F]$.

Let us consider for the moment only one of the faces of the octahedron. For representing the triangles after j levels of the subdivision process, we use (two-dimensional) barycentric coordinates. Let the vertices of the triangular face at hand have the coordinates $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$, where these coordinates correspond to the vertices of T_d given above in this order. For example, in T_1 we have $B = (1, 0, 0)$, $C = (0, 1, 0)$ and $A = (0, 0, 1)$ etc. The triangles T in $\mathcal{T}^j$ are uniquely determined by the face d and by their three vertices in barycentric coordinates. Observe that the triangles $T = [d; M_1, M_2, M_3]$ in $\mathcal{T}^j$ are either of type 1 or of type 2 with the following coordinates,

Type 1 triangle:

$$T = \Big[d;\, (\tfrac{p+1}{2^j}, \tfrac{m}{2^j}, \tfrac{k}{2^j}),\, (\tfrac{p}{2^j}, \tfrac{m+1}{2^j}, \tfrac{k}{2^j}),\, (\tfrac{p}{2^j}, \tfrac{m}{2^j}, \tfrac{k+1}{2^j})\Big] \quad p, m, k = 0 \ldots, 2^j - 1, \tag{1}$$

Type 2 triangle:

$$T = \Big[d;\, (\tfrac{p}{2^j}, \tfrac{m+1}{2^j}, \tfrac{k+1}{2^j}),\, (\tfrac{p+1}{2^j}, \tfrac{m}{2^j}, \tfrac{k+1}{2^j}),\, (\tfrac{p+1}{2^j}, \tfrac{m+1}{2^j}, \tfrac{k}{2^j})\Big] \quad p, m, k = 0 \dots, 2^j - 1, \qquad (2)$$

where in the first case $p + m + k = 2^j - 1$, and in the second case $p + m + k = 2^j - 2$. For example, the gray triangle in Figure 4 is of type 2, and its two direct neighbors in the row $r = 3$ are of type 1.

Let $\mathcal{T}_d^j \subset \mathcal{T}^j$ be the triangulation of one triangular face T_d. This triangulation $\mathcal{T}_d^j$ consists of 4^j triangles. In order to define a bijective mapping of $[d; M_1, M_2, M_3]$ in $\mathcal{T}^j$ to a one-dimensional index set, we first consider a mapping of the triangles to rows r and positions n in $\mathcal{T}_d^j$, i.e. a mapping $[d; M_1, M_2, M_3] \mapsto [d; r, n]$ as follows.

We say that a triangle $[d; M_1, M_2, M_3]$ is in the rth row of $\mathcal{T}_d^j$ if the z-coordinates of M_1, M_2 and M_3 (in the barycentric system) are all either $\frac{1}{2^j}(2^j - r)$ or $\frac{1}{2^j}(2^j - r - 1)$. Obviously, we have 2^j rows in the triangulation $\mathcal{T}_d^j$, and the rth row contains $2r - 1$ triangles, see Figure 4 for $j = 2$. Further, with n we denote the position of a triangle in the rth row, starting with the maximal x-coordinate in the barycentric coordinate system. For example, the gray triangle in Figure 4 is of type 2 and uniquely given by the coordinates $[d; (1/4, 1/2, 1/4), (1/4, 1/4, 1/2), (0, 1/2, 1/2)]$, or equivalently (with $r = 3$ and $n = 4$) by $[d; 3, 4]$.

Generally, we first determine the row and then the position of a triangle in the rth row by considering its distance from the vertex $(1, 0, 0)$. We observe that the triangle of type 1 in (1) is mapped to the row $r = 2^j - k$ and the position $n = 2(r - p) - 1$, and the triangle of type 2 in (2) is mapped to the row $r = 2^j - k$ and the position $n = 2(r - p) - 2$. Conversely, for a triangle $T = [d;\, r, n]$ we find:
If n is odd, then T is of type 1 with

$$k = 2^j - r, \quad p = r - \tfrac{n+1}{2}, \quad m = 2^j - 1 - k - p;$$

if n is even, then T is of type 2 with

$$k = 2^j - r, \quad p = r - \tfrac{n}{2} - 1, \quad m = 2^j - 2 - k - p.$$

With this preliminaries we are ready to determine the invertible mapping of triangles $T = [d;\, M_1, M_2, M_3]$ resp. $T = (d; r, n)$ in $\mathcal{T}^j$ to the one-dimensional index set indicated in Figure 1. We restrict ourselves to the faces $1, 2, 3, 4$ on the northern part of the octahedron, the other part can be done simply by reflection arguments.

Obviously, the rth row through all four northern triangular faces contains $4(2r - 1) = 8r - 4$ triangles (see e.g. Figure 1). Let now $T = [d;\, M_1, M_2, M_3]$ be an arbitrary triangle in $\mathcal{T}^j$ with $d \in \{1, 2, 3, 4\}$, and let $(d;\, r, n)$ be the (face, row, position) coordinates of this triangle. Then we determine the one-dimensional index $l \in \{1, \dots, 4^{j+1}\}$ of T as

$$l := 4(r - 1)^2 + (d - 1)(2r - 1) + n. \qquad (3)$$

Here we have used the fact that $4(r - 1)^2$ triangles belong to the first $r - 1$ rows of the triangulation $\mathcal{T}^j$, and $(d - 1)(2r - 1)$ triangles belong to the rth row in previous faces. So, starting at the North Pole, the one-dimensional index runs through the first row, then

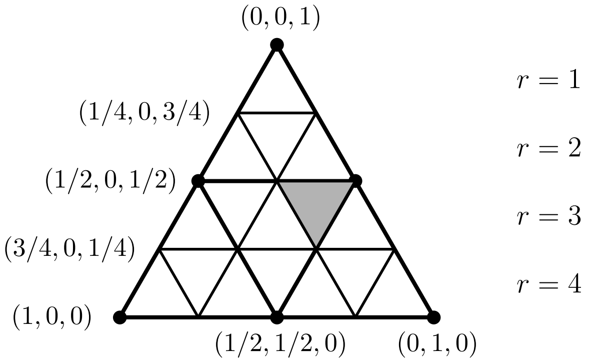

Figure 4. Illustration of the triangulation $\mathcal{T}_d^2$ on one face T_d of the octahedron with barycentric coordinates. The grey triangle is the 4th triangle in the third row.

through the second row etc., and it ends at the South Pole (in the 2^{j+1}th row, see Figure 1 (right)). Conversely, a triangle T indexed by $l \in \{1, \dots, 4^{j+1}\}$ in the one-dimensional index set J is in the rth row if $4(r-1)^2 < l \leq 4r^2$, i.e., we have

$$r := \left\lceil \frac{\sqrt{l}}{2} \right\rceil,$$

where $\lceil x \rceil$ denotes the smallest integer greater than x. Since each face has $2r-1$ triangles in the rth row, we find with $n_1 := l - 4(r-1)^2$ the face d of the triangle as

$$d := \left\lceil \frac{n_1}{2r-1} \right\rceil$$

and finally the position $n := n_1 - (d-1)(2r-1)$.

On the southern hemisphere, the one-dimensional mapping is continued with $l \in \{4^{j+1}+1, \dots, 2^{2j+3}\}$ by using reflection arguments, i.e., instead of l in (3) we take $2^{2j+3} - l + 1$.

Let us remark that this mapping $l = J([d; M_1, M_2, M_3])$ is only necessary as a preliminary step in order to relate the data vector correctly with the one-dimensional index set, and after finishing the EPWT, for correct projection of the obtained data onto the sphere.

4.2 Neighbors in the one-dimensional index set

During the transformation process using the EPWT, we work only with the one-dimensional index set and the corresponding vector of data. For this purpose, we need to determine the (one-dimensional) indices of the direct neighbors of a triangle $T_l \in \mathcal{T}^j$. We consider only the neighborhood with three neighbors, as also determined in Section 2.2., and the idea can be simply transferred to larger neighborhoods of T_l.

For $l = 1$, we find the neighborhood $\{2, 4, 6\}$ (see also Figure 1 (right)), and analogously, for the last index $l = 2^{2j+3}$ we obtain the neighborhood $\{2^{2j+3}-1, 2^{2j+3}-3, 2^{2j+3}-5\}$. For $2 \leq l \leq 2^{2j+3}-1$, two of the wanted neighbors are in the same row as l. If $l \leq 4^{j+1}$ and $l \neq 4\kappa^2$ (or $l > 4^{j+1}$ and $l \neq 2^{2j+3} - 4\kappa^2$) for some $\kappa \in \mathbb{N}$, then the two neighbors are immediately provided by $l-1$ and $l+1$. If $l = 4\kappa^2$ with $1 \le \kappa \le 2^j$, then the two neighbors of l are $l-1$ and $4(\kappa-1)^2+1$. Similarly, on the southern hemisphere, for $l = 2^{2j+3} - 4\kappa^2$ with $1 \le \kappa \le 2^j$, we have the neighbors $l-1$ and $2^{2j+3} - 4(\kappa+1)^2 + 1$.

The remaining third neighbor of l can be either in the previous or in the next row, depending on the type of the triangle T_l. We again use the (face, row, position) representation of the triangle T_l for determining this neighbor. We restrict ourselves to the case $l \le 4^{j+1}$. First we observe that T_l has the representation $(d; r, n)$ with

$$r = \left\lceil \frac{l}{2} \right\rceil, \; n_1 = l - 4(r-1)^2, \; d = \left\lceil \frac{n_1}{2r-1} \right\rceil, \; n := n_1 - (d-1)(2r-1),$$

see Section 4.1. For odd n, the triangle T_l is of type 1 and its third neighbor is in the next row. For $r < 2^j$, this neighbor is determined by $(d, r+1, n+1)$, i.e., by the one-dimensional index

$$4r^2 + (d-1)(2r+1) + n + 1 = l + 8r + 2d - 5.$$

In the case $r = 2^j$ the third neighbor belongs to the southern hemisphere, and has the index $l + 4(2^{j+1}-1)$ since $4(2^{j+1}-1)$ triangles are contained in the 2^jth row and also in the (2^j+1)th row of $\mathcal{T}^j$.

For even n, the triangle T_l is of type 2 and its neighbor is in the previous row. It is determined by $(d, r-1, n-1)$ i.e. by the one-dimensional index

$$4(r-2)^2 + (d-1)(2r-3) + n - 1 = l - 8r - 2d + 13.$$

4.3 Numerical experiments

To illustrate the efficiency of our method, we took the data set *topo*, and we considered the regular octahedron with triangulation $\mathcal{T}_6$, containing 32768 triangles. The approximation $\mathbf{f}^6$ on $\mathcal{U}_6$ is represented in Figure 5 (left). Applying 4 levels of the EPWT, we obtain 2048 scaling coefficients and 30720 wavelet coefficients. We have used different thresholds to the wavelet coefficients, see Table 1. Figure 5 (right) shows the reconstructed data $\widetilde{\mathbf{f}}^6$ using only 2265 wavelet coefficients. For comparison of the reconstructed data $\widetilde{\mathbf{f}}^6$ with the original data $\mathbf{f}^6$, we measured the SNR given as

$$SNR = 20 \cdot \log_{10} \frac{\|\mathbf{f}^6 - \mathrm{mean}(\mathbf{f}^6)\|_2}{\|\mathbf{f}^6 - \widetilde{\mathbf{f}}^6\|_2}.$$

The results are contained in Table 1, where the mean of $\mathbf{f}^6$ is -2329.

5 Conclusions

In this paper, we have proposed a new method for sparse representation of data on the sphere using the concept of EPWT. This method is particularly efficient for compression of data sets containing texture components, as e.g. topographical data. As in the two-dimensional case, the efficiency of the EPWT depends strongly on the "cost of adaptivity", i.e. the cost for storing the path vectors $\mathbf{p}^j$ at each level of the transform. In order to

threshold	number of remaining wavelet coeff.	percent of remaining wavelet coefficients	SNR
1	27732	90.27 %	84.72
100	14185	46.17 %	38.59
500	5230	17.02 %	25.30
1000	3313	10.78 %	21.17
1500	2699	8.78 %	19.18
2000	2402	7.81 %	17.89
2500	2265	7.37 %	17.10

Table 1: Compression results for the data set *topo*.

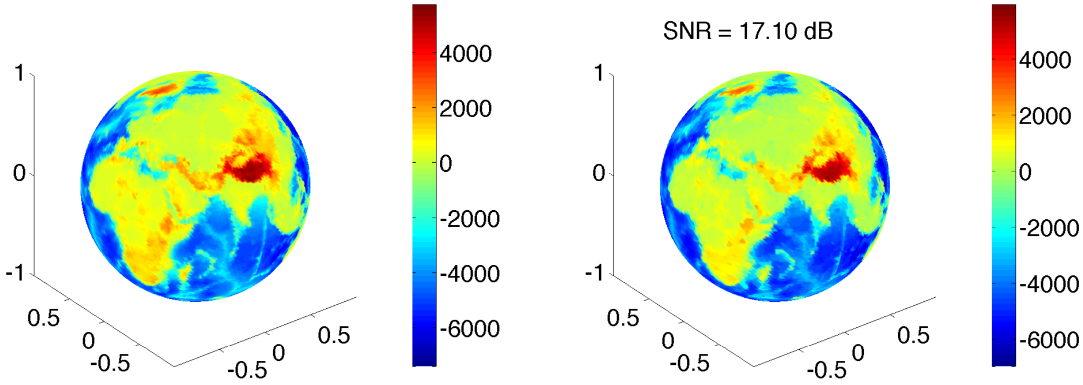

Figure 5. Approximation $\mathbf{f}^6$ at level 6 of the original data set *topo* and the compressed version $\widetilde{\mathbf{f}}^6$ with threshold 2500.

reduce these costs, we have proposed a so-called relaxed EPWT in Plonka (2009) using the idea of "favorite directions". This approach can be implemented on the sphere as well.

Furthermore, we want to emphasize that the EPWT can also be applied to planar triangulations and to triangulations of parts of the sphere. We only need a one-dimensional indexing of all triangles of the triangulation and a definition of neighborhood of each triangle. The application of the EPWT on the sphere is not restricted to a triangular grid. In a forthcoming paper, we will also consider the performance of the EPWT for other grids like HEALPIX and present further numerical experiments showing the remarkable efficiency of this approach.

Acknowledgement. This research in this paper is supported by the project 436 RUM 113/31/0-1 of the German Research Foundation (DFG). This is gratefully acknowledged.

References

Claypoole RL, Davis GM, Sweldens W, Baraniuk RG (2003) Nonlinear wavelet transforms for image coding via lifting. IEEE Trans Image Process 12: 1449-1459

Cohen A, Matei B (2001) Compact representation of images by edge adapted multiscale transforms. In Proc IEEE Int Conf on Image Proc (ICIP), Thessaloniki, 8-11

Dekel S, Leviatan D (2006) Adaptive multivariate approximation using binary space partitions and geometric wavelets. SIAM J Numer Anal 43: 707-732

Ding W, Wu F, Wu X, Li S, Li H (2007) Adaptive directional lifting-based wavelet transform for image coding. IEEE Trans Image Process 16: 416-427

Donoho DL (1999) Wedgelets: Nearly minimax estimation of edges. Ann Stat 27: 859-897

Mallat S (2009) Geometrical grouplets. Appl Comput Harmon Anal 26: 161-180

Plonka G (2009) The easy path wavelet transform: A new adaptive wavelet transform for sparse representation of two-dimensional data. Multiscale Model Simul 7: 1474-1496

Roşca D (2005a) Haar wavelets on spherical triangulations. In: Dodgson NA, Floater MS, Sabin MA (eds) Advances in Multiresolution for Geometric Modelling. Springer, 405-417

Roşca D (2005b) Locally supported rational spline wavelets on a sphere. Math Comput 74: 1803-1829

Roşca D (2009) On a norm equivalence in $L^2(\mathbb{S}^2)$. Result Math: to appear

Schröder P, Sweldens W (1995) Spherical wavelets: efficiently representing functions on the sphere. In: ACM SIGGRAPH, New York, 161-172

Shukla R, Dragotti PL, Do MN, Vetterli M (2005) Rate-distortion optimized tree structured compression algorithms for piecewise smooth images. IEEE Trans Image Process 14: 343-359