

Sparse Data Representation on the Sphere using the Easy Path Wavelet Transform

Gerlind Plonka ⁽¹⁾ and Daniela Roșca ⁽²⁾

(1) Department of Mathematics, University of Duisburg-Essen, Campus Duisburg, 47048 Duisburg, Germany.

(2) Department of Mathematics, Technical University of Cluj-Napoca, 400020 Cluj-Napoca, Romania.

gerlind.plonka@uni-due.de, Daniela.Rosca@math.utcluj.ro

Abstract:

In this paper we consider the Easy Path Wavelet Transform (EPWT) on spherical triangulations. The EPWT has been introduced in [7] in order to obtain sparse image representations. 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 one-dimensional discrete wavelet transform (DWT), orthogonal or biorthogonal, can be applied.

1. Introduction

One important problem in data analysis 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.

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, that try to exploit the local correlations of the data, have been developed (see e.g. [1, 2, 3, 4, 5, 6, 7, 10]).

We will focus on the EPWT recently introduced in [7] 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 by Roșca [8, 9] 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 [8, 9] for the construction of Haar wavelets and of locally supported rational spline wavelets on the sphere.

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 to its corresponding (one-dimensional) index.

In the first step we select a path through the complete index set in such a way that data points associated to neighbor indices in the path are strongly correlated. For this purpose, for each index we choose “the best” neighbor index that has not been used in the path yet, 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.

2. Spatial and spherical triangulations

Consider the sphere $\mathbb{S}^2 = \{\mathbf{x} \in \mathbb{R}^3, \|\mathbf{x}\|_2 = 1\}$ and let Π be a convex polyhedron with triangular faces, containing O inside. For example we can take an icosahedron, a cube with triangulated faces, an octahedron, etc. The boundary of the polyhedron will be denoted by Ω . We denote by $\mathcal{T}^0 = \{T_1, \dots, T_M\}$ the set of faces of Π . For each triangle $T \in \mathcal{T}^0$ we take the mid-points of its edges and construct four triangles of equal area, as in Figure 1. All these small triangles will form a refined triangulation of \mathcal{T}^0 , denoted \mathcal{T}^1 . Continuing the refinement process in the same manner, 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. For application of the EPWT we will need a one-dimensional index set $J = J^j$ for the triangles in \mathcal{T}^j . Using the octahedron, this one-dimensional index set J can be as in Figure 1 (right). Observe that for the octahedron the number of triangles at the j th level is given by $\#J = \#\mathcal{T}^j = 2^{2j+3}$. In order to obtain a spherical triangulation, for the given

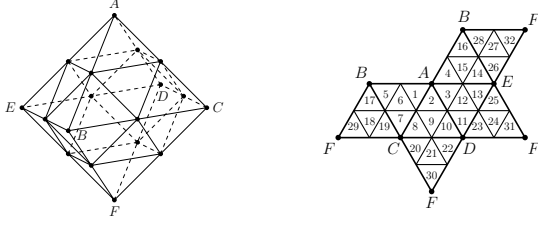


Figure 1: Illustration of the octahedron with triangulation \mathcal{T}^1 (left) and a fold apart version of the octahedron on the plane, with a one-dimensional indexing of all triangles.

polyhedron Π we define the radial projection $p : \Omega \rightarrow \mathbb{S}^2$,
 $p(x, y, z) = (x^2 + y^2 + z^2)^{-1/2} \cdot (x, y, z)$, $(x, y, z) \in \Omega$.

The set $\mathcal{U}^j = \{U = p(T), T \in \mathcal{T}^j\}$ will be a triangulation of the sphere \mathbb{S}^2 . For indexing the spherical triangles in \mathcal{U}^j , we use the same index set J as for the triangulation \mathcal{T}^j of the polyhedron.

3. 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 need some definitions and notations.

Let us assume that a fixed refined spherical triangulation \mathcal{U}^j is given. Let J be a one-dimensional index set for the spherical triangles in \mathcal{U}^j .

We define a *neighborhood* of an index $\nu \in J$ as

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

Hence, each index $\nu \in J$ has exactly three neighbors. One may also use a bigger neighborhood, e.g. $\mathcal{N}(\nu) = \{\mu \in J \setminus \{\nu\} : T_\mu \text{ and } T_\nu \text{ have a common edge or a common vertex}\}$, in which case each index has 12 neighbors.

We also need a definition of neighborhood of subsets of an index set. We shall consider disjoint *partitions* of J of the form $\{J_1, J_2, \dots, J_r\}$, 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 the indices $l \in J_\nu$ and $l_1 \in J_\mu$ such that $l \in \mathcal{N}(l_1)$. We consider a function f being piecewise constant on the triangles of \mathcal{U}^j , i.e., we identify each spherical triangle in \mathcal{U}^j with a 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 we apply a one-dimensional wavelet transform along these path vectors. Any orthogonal or biorthogonal one-dimensional wavelet transform can be used here.

4. Description of the EPWT

In this section we give a summary of the idea of the EPWT, described in more details in [7]. 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 $L \in \mathbb{N}$. Then we will be able to apply L levels of the EPWT. For the considered octahedron we have $N = 2^{2j+3}$.

Decomposition

First level

We first determine a complete path vector \mathbf{p}^L through the index set $J = \{1, 2, \dots, N\}$ and then apply a suitable discrete one-dimensional (periodic) wavelet transform to the function values $\mathbf{f}^L = (\mathbf{f}^L(j))_{j \in J}$ along the path \mathbf{p}^L . We start with $\mathbf{p}^L(1) := 1$. Next, for $\mathbf{p}^L(2)$ we take

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

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 described above, we obtain a pathway such that the absolute differences between neighboring function values $\mathbf{f}^L(l)$ along the path are as small as possible. In general, for a given the index $\mathbf{p}^L(l)$, $1 \leq l \leq N - 1$, the next value $\mathbf{p}^L(l + 1)$ is defined by

$$\mathbf{p}^L(l + 1) := \underset{k}{\operatorname{argmin}} \{|\mathbf{f}^L(\mathbf{p}^L(l)) - \mathbf{f}^L(k)|, k \in \mathcal{N}(\mathbf{p}^L(l)) \setminus \{\mathbf{p}^L(\nu), \nu = 1, \dots, l\}\}.$$

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 for more than one index. In this case, one may fix favorite directions in order to determine a unique pathway.

Another situation which can occur during the procedure is that all indices in the neighborhood of an index $\mathbf{p}^L(l)$ have already been used in the path \mathbf{p}^L . In this case 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 , which have not been taken yet in \mathbf{p}^L . There are different possibilities for finding a suitable next index. One simple choice is to take 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

$$\mathbf{p}^L(l + 1) = \underset{k}{\operatorname{argmin}} \{|\mathbf{f}^L(\mathbf{p}^L(l)) - \mathbf{f}^L(k)|, k \in J \setminus \{\mathbf{p}^L(\nu), \nu = 1, \dots, l\}\}.$$

By proceeding in this manner, we finally obtain a path vector $\mathbf{p}^L \in \mathbb{Z}^N$, which is a permutation of $(1, 2, \dots, N)$.

After having constructed the path \mathbf{p}^L , we apply one level of the 1-D Haar DWT (or any other orthogonal or biorthogonal periodic DWT) to the vector of function values $(\mathbf{f}^L(\mathbf{p}^L(l)))_{l=1}^N$ along the path \mathbf{p}^L . We obtain 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 further proceed with the low-pass vector \mathbf{f}^{L-1} at the second level.

Further levels

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

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

with the corresponding function values $(\mathbf{f}^{L-j}(l))_{l=1}^{N/2^j}$. In particular, the index sets at the second level are $J_l^{L-1} := \{\mathbf{p}^L(2l-1), \mathbf{p}^L(2l)\}$, $l = 1, \dots, N/2$, determining a partition of J .

We repeat the procedure described in the first step, but replacing the single indices with the new index sets J_l^{L-j} , and the corresponding function values with the smoothed function values $\mathbf{f}^{L-j}(l)$.

The new path vector $\mathbf{p}^{L-j} \in \mathbb{Z}^{N/2^j}$ should now be a permutation of $(1, 2, \dots, N/2^j)$. We start again with the first index set J_1^{L-j} , i.e., $\mathbf{p}^{L-j}(1) = 1$. Having already found $\mathbf{p}^{L-j}(l)$, $1 \leq l \leq N/2^j - 1$, we determine the next value $\mathbf{p}^{L-j}(l+1)$ as

$$\mathbf{p}^{L-j}(l+1) = \underset{k}{\operatorname{argmin}} \{|\mathbf{f}^{L-j}(\mathbf{p}^{L-j}(l)) - \mathbf{f}^{L-j}(k)|, \\ J_k^{L-j} \in \mathcal{N}(J_{\mathbf{p}^{L-j}(l)}^{L-j}) \setminus \{\mathbf{p}^{L-j}(\nu), \nu = 1, \dots, l\}\}.$$

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

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

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

and the vector determining the paths at each iteration step

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

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 , $j = 0, \dots, L-1$ and obtain the vectors $\tilde{\mathbf{g}}^j$.

Reconstruction

The reconstruction of \mathbf{f}^L from $\tilde{\mathbf{g}} = (\mathbf{f}^0, \tilde{\mathbf{g}}^0, \tilde{\mathbf{g}}^1, \dots, \tilde{\mathbf{g}}^{L-1})$ and \mathbf{p} is given as follows.

$$\tilde{\mathbf{f}}^0 = \mathbf{f}^0;$$

For $j = 0$ to $L-1$

- Apply the inverse DWT to the vector $(\tilde{\mathbf{f}}^j, \tilde{\mathbf{g}}^j) \in \mathbb{R}^{r2^j}$ in order to obtain $\tilde{\mathbf{f}}_p^{j+1} \in \mathbb{R}^{r2^{j+1}}$.

- Apply the permutation $\tilde{\mathbf{f}}^{j+1}(\mathbf{p}^{j+1}(k)) = \tilde{\mathbf{f}}_p^{j+1}(k)$, for $k = 1, \dots, r2^{j+1}$.

5. Example

We illustrate the simple 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

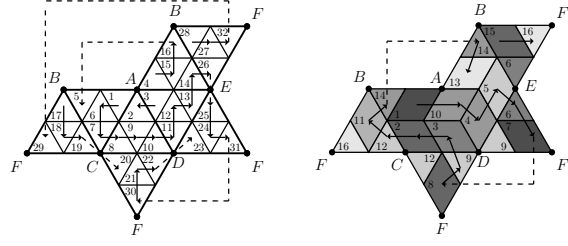


Figure 2. Illustration of first path through the triangulation T^1 of the octahedron (left) and of the low-pass part after the first level of EPWT with Haar DWT (right). Index sets at the second level are illustrated by different gray values, and path vectors are represented by arrows.

sphere, where each function value corresponds to a spherical triangle that has been obtained by radial projection of the triangulated octahedron in Figure 1 (left). The values are given as a vector $\mathbf{f} = \mathbf{f}^5$ of length 32, corresponding to the one-dimensional indexing of the triangles in Figure 1 (right),

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

Starting with the index 1, with the function value 0.4492, we determine the first path vector. This index has the three neighbors 2, 4, and 6, with the 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 4 we obtain

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

where the interruptions in the path are indicated by semi-colons. This path has four interruptions and is illustrated by arrows in Figure 2 (left). An application of the Haar DWT (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

$$\mathbf{f}^4 = (0.4512, 0.4219, 0.4334, 0.4219, 0.4238, 0.4219, \\ 0.4219, 0.4200, 0.4140, 0.4238, 0.4219, 0.4336, 0.4199, \\ 0.4141, 0.4336, 0.4434),$$

and the wavelet coefficients

$$\mathbf{g}^4 = (-0.0020, -0.0039, -0.0042, 0., -0.0020, \\ -0.0039, 0., 0.0058, -0.0118, 0.0020, 0., -0.0039, \\ 0.0176, 0., -0.0195, 0.0058).$$

We now proceed to the second level. For the smoothed vector of function values \mathbf{f}^4 corresponding to the 16 index

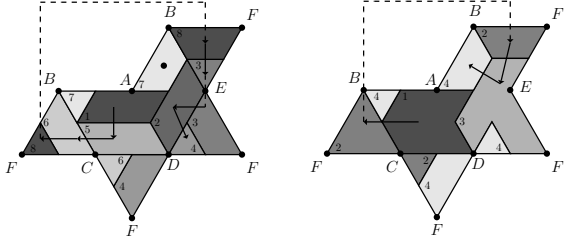


Figure 3. Illustration of the third and fourth paths.

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). An application of the Haar DWT along \mathbf{p}^4 gives

$$\mathbf{f}^3 = (0.4375, 0.4229, 0.4219, 0.4170, 0.4276, 0.4278, 0.4170, 0.4385),$$

$$\mathbf{g}^3 = (0.0136, -0.0010, 0., 0.0030, 0.0057, 0.0058, 0.0029, -0.0049).$$

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

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

At 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

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

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

6. Numerical experiments

To illustrate the efficiency of our method, we took the dataset *topo* and we considered the regular octahedron with triangulation \mathcal{T}_6 , containing 32768 triangles. The approximation \mathbf{f}^6 at level 6 is represented in Figure 4. We applied the EPWT with different thresholds, obtaining the compressed vector $\tilde{\mathbf{f}}^6$, and we measured the SNR given as

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

threshold	number of remaining wavelet coeff.	l^2 -norm of error	SNR
1	27732	26.4031	84.72
100	14185	5.34e+03	38.59
500	5230	2.47e+04	25.30
1000	3313	3.97e+04	21.17
1500	2699	5.00e+04	19.18
2000	2402	5.79e+04	17.89
2500	2265	6.35e+04	17.10

Table 1: Compression results for the dataset *topo*.

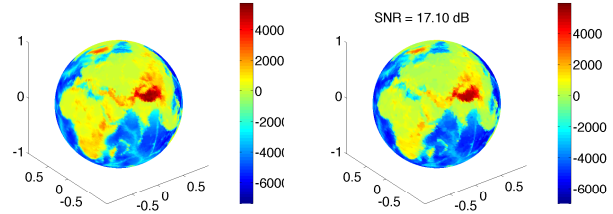


Figure 4. Approximation \mathbf{f}^6 at level 6 of the original dataset *topo* and the compressed version $\tilde{\mathbf{f}}^6$ with threshold 2500.

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

Acknowledgments

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:

- [1] R.L. Claypoole, G.M. Davis, W. Sweldens, and R.G. Baraniuk. Nonlinear wavelet transforms for image coding via lifting. *IEEE Trans. Image Process.* 12:1449–1459, 2003.
- [2] A. Cohen and B. Matei. Compact representation of images by edge adapted multiscale transforms. In *Proc. IEEE Int. Conf. on Image Process. (ICIP)*, Thessaloniki, pages 8–11, 2001.
- [3] S. Dekel and D. Leviatan. Adaptive multivariate approximation using binary space partitions and geometric wavelets. *SIAM J. Numer. Anal.* 43:707–732, 2006.
- [4] W. Ding, F. Wu, X. Wu, S. Li, and H. Li. Adaptive directional lifting-based wavelet transform for image coding. *IEEE Trans. Image Process.* 16:416–427, 2007.
- [5] D.L. Donoho. Wedgelets: Nearly minimax estimation of edges. *Ann. Stat.* 27:859–897, 1999.
- [6] S. Mallat. Geometrical grouplets. *Appl. Comput. Harmon. Anal.*, 26 (2): 143–290, 2009.
- [7] G. Plonka. The easy path wavelet transform: A new adaptive wavelet transform for sparse representation of two-dimensional data. *Multiscale Model. Simul.* 7:1474–1496, 2009.
- [8] D. Roşca. Haar wavelets on spherical triangulations. In Dodgson, N.A., Floater, M.S., Sabin, M.A., editors, *Advances in Multiresolution for Geometric Modelling*, Springer, pages 405–417, 2005.
- [9] D. Roşca. Locally supported rational spline wavelets on a sphere. *Math. Comput.* 74:1803–1829, 2005.
- [10] R. Shukla, P.L. Dragotti, M.N. Do, and M. Vetterli. Rate-distortion optimized tree structured compression algorithms for piecewise smooth images. *IEEE Trans. Image Process.* 14:343–359, 2005.