

Wavelet shrinkage on paths for denoising of scattered data

Dennis Heinen and Gerlind Plonka

Dedicated to Werner Haußmann in memoriam

Abstract. We propose a new algorithm for denoising of multivariate function values given at scattered points in \mathbb{R}^d . The method is based on the one-dimensional wavelet transform that is applied along suitably chosen path vectors at each transform level. The idea can be seen as a generalization of the relaxed easy path wavelet transform in [13] to the case of multivariate scattered data. The choice of the path vectors is crucial for the success of the algorithm. We propose two adaptive path constructions that take the distribution of the scattered points as well as the corresponding function values into account. Further, we present some theoretical results on the wavelet transform along path vectors in order to indicate that the wavelet shrinkage along path vectors can really remove noise. The numerical results show the efficiency of the proposed denoising method.

Mathematics Subject Classification (2000). Primary 42C40, 65T60; Secondary 68U10.

Keywords. scattered data denoising, wavelet shrinkage, path vectors, easy path wavelet transform.

1. Introduction

Within the last years, wavelet threshold methods have been shown to be a suitable tool for denoising of functions and images. In particular, using a translation-invariant filter bank, the visual artifacts in the neighborhood of discontinuities are well suppressed [6]. The basic idea of wavelet denoising methods is a suitable separation of frequencies of a given noisy signal f . Supposing that the noise corresponds to wavelet coefficients with a small amplitude, the application of a thresholding procedure to the wavelet expansion of f removes it from the signal. Compared to other approaches to

image/signal denoising (as nonlinear diffusion or variational methods), one advantage of the wavelet denoising method is its numerical efficiency.

However, the traditional (tensor-product) wavelet transform only applies for equidistant grids. There have been several attempts to generalize the wavelet transform to functions that are sampled on non-equidistant or scattered points. In the one-dimensional case, some of these methods apply a certain approximation in a first step in order to get back the equidistant design situation [1, 3, 10, 12]. Other approaches that particularly apply in two or higher dimensions are based on the construction of second generation wavelets [17] by the lifting scheme, see e.g. [2, 7, 11, 19]. These wavelet constructions adaptively depend on the scattered points and the corresponding function values and therefore lose much of the simplicity and efficiency of the traditional wavelet transform.

In [13], the easy path wavelet transform (EPWT) has been introduced by one of the authors for sparse image approximation. The EPWT employs the one-dimensional wavelet transform along path vectors through the image values. The path vectors contain all indices (i, j) corresponding to the image values $f(i, j)$ and are determined in a way such that neighboring indices in the path are also neighboring in the two-dimensional grid and that the corresponding image values are well correlated. Obviously, the EPWT can be simply transferred to the setting of scattered data $x_j \in \mathbb{R}^d$ with corresponding function values $f(x_j) \in \mathbb{R}$, where we have just to generalize the notion of a neighborhood for x_j . A similar idea has been already applied in [14] for data on the sphere. The recently proposed generalized tree-based wavelet transform (GTBWT) [16] is closely related to the EPWT [13] and generalizes the Haar-like transform in [9]. In particular, the tree-based wavelet transform has been combined with a suitable sub-image averaging scheme for image denoising.

In this paper, we want to propose a new adaptive wavelet threshold scheme for scattered data denoising. The basic idea of our scheme is very similar to the EPWT, namely to employ the usual one-dimensional wavelet transform along suitable path vectors at each level. But we need to determine the path vectors differently in order to obtain good denoising results. Further, for resembling the cycle spinning method [6], we apply the wavelet threshold scheme several times along different path vectors and compute the average of the results.

The paper is organized as follows. In Section 2 we describe the general denoising algorithm along given path vectors. In Section 3 we propose two different methods for determining the path vectors, a deterministic and a random path construction. The two methods are both adaptive; the choice of the next path component adaptively depends on the spatial distance of the involved points x_j and the distance of the involved function values $f(x_j)$ (resp. the low pass values at further levels of the wavelet transform). In Section 4, we explore some theoretical results that support our approach for scattered data denoising. Finally, in Section 5 we present numerical results

of our data denoising method in the two-dimensional case. For the special case of images we also present some comparisons with other image denoising methods.

2. Description of the algorithm

Let $\Gamma = \{x_1, \dots, x_N\}$ be the data set with scattered data points $x_j \in \mathbb{R}^d$, and let $f : \mathbb{R}^d \mapsto \mathbb{R}$ be a function sampled on $\Gamma \subset \Omega$, where Ω is a connected subset of \mathbb{R}^d consisting of a finite union of convex domains. This means, the measured values $\tilde{f}(x_j)$ are given, where we assume that

$$\tilde{f}(x_j) = f(x_j) + z_j,$$

and z_j denotes additive Gaussian noise with zero mean and an (unknown) variance σ^2 . For the distribution of the scattered data set Γ , we suppose quasi-uniformity, i.e., considering the maximal density

$$\delta(\Gamma) := \max_{x \in \Omega} \min_{k=1, \dots, N} |x - x_k|$$

and the minimal spacing

$$\mu(\Gamma) := \min_{\substack{x_j, x_k \in \Gamma \\ j \neq k}} |x_j - x_k|,$$

we assume that $\delta(\Gamma) < C \cdot \mu(\Gamma)$, with a constant C independent of Ω and Γ . Further, we suppose that f is a piecewise smooth function.

Now, the basic idea of the new algorithm is the application of the classical wavelet shrinkage procedure along one-dimensional path vectors thereby generalizing the easy path wavelet transform (EPWT), see [13].

The EPWT was introduced for image approximation. Therefore, the path construction used in [13] is based on the strong correlation of function values corresponding to neighboring points in the path vector. A new point in the path vector is taken from the set of neighbor points, such that the difference of the corresponding function values is minimal. In case of noisy data, this approach for the path vector construction needs to be changed since the correlation of function values is now influenced by noise. In our algorithm, the construction of suitable path vectors turns out to be crucial for a good denoising performance.

Similarly to [9] and [16], we resemble the "cycle spinning" method [6] in order to improve the denoising procedure. For image denoising, it is usual to apply the tensor product wavelet shrinkage procedure not only to the noisy image itself but also to all images obtained by up to 7 cyclic shifts in x - and in y -direction. After application of the wavelet transform to each of the 64 images obtained in this way, an averaging is applied that greatly improves the denoising result. In our approach, we will just apply the path wavelet shrinkage procedure repeatedly along different path vectors and then average the result.

We summarize the complete algorithm before describing the path construction in more detail.

Algorithm 2.1.

Given: the point set $\Gamma = \{x_1, \dots, x_N\} = \{x_1^0, \dots, x_N^0\} = \Gamma^0 \subset \mathbb{R}^d$,

$$c_j^0 := \tilde{f}(x_j) = f(x_j) + z_j, \quad j = 1, \dots, N, \quad N = 2^t,$$

a biorthogonal wavelet filterbank with decomposition filters \tilde{h}, \tilde{g} and reconstruction filters h, g

Iteration: Perform the following 4 steps for $l = 1, 2, \dots, L$ with $L < t$:

1. Find a suitable path vector $p^l \in \mathbb{R}^{N/2^l}$ consisting of a permutation of $\{1, \dots, N/2^l\}$ that describes a fixed order of points $x_{p^l(j)}^l$ and of the corresponding function values $c_{p^l(j)}^l$.
2. Apply the (periodic) low-pass filter \tilde{h} to $(c_{p^l(j)}^l)_{j=1}^N$ followed by downsampling by two to obtain the low-pass data $(c_j^{l+1})_{j=1}^{N/2^{l+1}}$. Apply the (periodic) high-pass filter \tilde{g} to $(c_{p^l(j)}^l)_{j=1}^N$ followed by downsampling by two to obtain the vector of wavelet coefficients $(d_j^{l+1})_{j=1}^{N/2^{l+1}}$.
3. Apply the normalized low-pass filter $\frac{1}{\sqrt{2}}\tilde{h}$ (such that $\frac{1}{\sqrt{2}}\sum_{l \in \mathbb{Z}} \tilde{h}_l = 1$) to the point vector $(x_{p^l(j)}^l)_{j=1}^{N/2^l}$ (i.e., separately to all d components), followed by downsampling by two to obtain a new vector of scattered points $(x_j^{l+1})_{j=1}^{N/2^{l+1}}$.
Determine the new point set $\Gamma^{l+1} := \{x_1^{l+1}, \dots, x_{N/2^{l+1}}^{l+1}\}$.

4. Apply a threshold procedure to the wavelet coefficients $(d_j^{l+1})_{j=1}^{N/2^{l+1}}$ to find

$$\tilde{d}_j^l = T_\theta(d_j^l) = \begin{cases} d_j^l & \text{if } |d_j^l| \geq \theta, \\ 0 & \text{if } |d_j^l| < \theta \end{cases}$$

with a predefined threshold parameter $\theta > 0$.

Reconstruct the values $f(x_j)$ by the following iteration, where $(\tilde{c}_j^L)_{j=1}^{N/2^L} := (c_j^L)_{j=1}^{N/2^L}$.

Iteration: Perform the following three steps for $l = L, L-1, \dots, 1$:

5. Apply an upsampling by two first and then the low-pass filter h to $(\tilde{c}_j^l)_{j=1}^{N/2^l}$.
6. Apply an upsampling by two first and then the high-pass filter g to $(\tilde{d}_j^l)_{j=1}^{N/2^l}$.
7. Add the results of the previous two steps to obtain $(\tilde{c}_{p^{l-1}(j)}^{l-1})_{j=1}^{N/2^{l-1}}$ and apply the reverse permutation to get $(\tilde{c}_j^{l-1})_{j=1}^{N/2^{l-1}}$.

Output: $(\tilde{c}_j^0)_{j=1}^N$ the smoothed function values at scattered points $x_j \in \Gamma$.

As already remarked before, analogously to the cycle spinning approach, we apply the above algorithm several times and average the results in order to improve the denoising method.

Remark 2.2.

1. Observe that, similarly as for the EPWT in [13], the considered algorithm is usually not just a one-dimensional transform since the path vectors change at each level of the wavelet transform. This point is crucial for the performance of the denoising method. In particular, the set of scattered data points related to the computed low-pass values changes at each level according to step 3 of the algorithm. For a one-dimensional wavelet transform at equidistant points x_j on a line, step 3 of the algorithm just transforms the point set again into a (scaled) equidistant point set of half length on the line and needs therefore not to be considered.

2. In our numerical experiments for bivariate data, we have also obtained satisfying results by omitting step 3 of Algorithm 2.1 and by relating the low-pass data $(c_j^l)_{j=1}^{N/2^l}$ to a suitable subset of Γ . Here we take $\Gamma^1 := \{x_j^1 := x_{p^1(2j-1)}^0 : j = 1, \dots, N/2\} \subset \Gamma$, and generally

$$\Gamma^{l+1} := \{x_j^{l+1} := x_{p^{l+1}(2j-1)}^l : j = 1, \dots, N/2^{l+1}\} \subset \Gamma.$$

3. It is possible to apply other shrinkage functions in step 4 of the algorithm as e.g. a firm shrinkage [8].

3. Construction of path vectors

The main challenge in the above denoising algorithm is to construct path vectors through the point sets Γ^l . As we will see in the sequel, the choice of the path vectors is crucial for the success of the denoising algorithm. For the path construction, we have to answer the following questions:

- How should one determine the neighborhood of scattered points as a tool for determination of the path vectors?
- How should one determine the distance between scattered points and between the corresponding function values? How we have to weight the distance $\|x_j^l - x_k^l\|_2$ and the difference of corresponding function/low-pass values $|c_j^l - c_k^l|$?
- Should the path vectors be chosen in a deterministic way or randomly, based on the point sets Γ^l and the corresponding data c_j^l ?

In the remainder of this section, we shall propose two path vector constructions that we will use in our numerical examples.

3.1. Adaptive deterministic path construction

In \mathbb{R}^d , let us apply a neighborhood definition of the form

$$N(x_j^l) = \{x_k^l \in \Gamma^l : \|x_j^l - x_k^l\|_2 \leq 2^{l/d} \cdot C_1, j \neq k\}, \quad (3.1)$$

where C_1 depends on the distribution of the original point set Γ . Observe that the path vector at level l of the wavelet transform is a permutation of the indices of the points $\{x_j^l : j = 1, \dots, N/2^l\}$. In particular, each point has

to occur exactly once in the path vector. In our numerical examples for $d = 2$, we take C_1 such that $N(x_j^0)$ with $x_j^0 := x_j \in \Gamma$ contains at least 8 entries for each $j \in \{1, \dots, N\}$. The path vectors at the level l of the wavelet transform are determined by the following procedure.

Algorithm 3.1. (Deterministic path construction)

Given: the point set $\Gamma^l = \{x_1^l, \dots, x_{N/2^l}^l\} \subset \mathbb{R}^d$,
low pass values c_j^l , $j = 1, \dots, N/2^l$, $N = 2^t$.

Choose the first path component $p^l(1)$ randomly from $\{1, \dots, N/2^l\}$.

Iteration: Perform the following steps for $k = 1, \dots, N/2^l - 1$:

1. For a fixed shrinkage parameter θ , compute the subset

$$N_\theta(x_{p^l(k)}^l) \subset \tilde{N}(x_{p^l(k)}^l) := \{x_r^l \in N(x_{p^l(k)}^l) : r \notin \{p^l(1), \dots, p^l(k)\}\}$$

of points that have not been used in the path vector yet and whose corresponding function/low-pass values differ at most by θ , i.e.,

$$N_\theta(x_{p^l(k)}^l) := \{x_r^l \in \tilde{N}(x_{p^l(k)}^l) : |c_{p^l(k)}^l - c_r^l| \leq \theta\}. \quad (3.2)$$

2. Choose the next component in the path vector as follows.

- If $N_\theta(x_{p^l(k)}^l)$ is not empty, then choose the next path component $p^l(k+1)$ such that

$$x_{p^l(k+1)}^l = \operatorname{argmin}_{x \in N_\theta(x_{p^l(k)}^l)} \left| \|x_{p^l(k)}^l - x_{p^l(k-1)}^l\|_2 - \|x_{p^l(k)}^l - x\|_2 \right|, \quad (3.3)$$

where we put $\|x_{p^l(1)}^l - x_{p^l(0)}^l\|_2 := 0$ for $k = 1$. For $k > 1$, one may alternatively choose $p^l(k+1)$ such that

$$x_{p^l(k+1)}^l = \operatorname{argmax}_{x \in N_\theta(x_{p^l(k)}^l)} \frac{\langle x_{p^l(k-1)}^l - x_{p^l(k)}^l, x_{p^l(k)}^l - x \rangle}{\|x_{p^l(k-1)}^l - x_{p^l(k)}^l\|_2 \cdot \|x_{p^l(k)}^l - x\|_2}. \quad (3.4)$$

- If $N_\theta(x_{p^l(k)}^l)$ is the empty set, we randomly choose the next path component from $\tilde{N}(x_{p^l(k)}^l)$.

If $\tilde{N}(x_{p^l(k)}^l)$ is also empty, we choose the next index in the path vector randomly from the remaining indices of the point set

$$\tilde{N}_{k,l} := \{x_r^l : r \notin \{p^l(1), \dots, p^l(k)\}\},$$

or, if possible, from the subset

$$\tilde{N}_{k,l,\theta} := \{x_r^l \in \tilde{N}_{k,l} : |c_{p^l(k)}^l - c_r^l| \leq \theta\} \subset \tilde{N}_{k,l}.$$

Output: path vector $p^l = (p^l(k))_{k=1}^{N/2^l}$.

Remark 3.2.

The parameter θ that has to be chosen for determining the neighborhood $N_\theta(x_{p^l(k)}^l)$ can be taken of the same size as the thresholding parameter for the wavelet transform in Algorithm 2.1.

Determining the $p^l(k+1)$ by (3.3) forces that scattered points corresponding to neighboring components in the path have similar distance, while (3.4) ensures the path to be continued in a similar direction as before, i.e., the angle between $x_{p^l(k-1)} - x_{p^l(k)}$ and $x_{p^l(k)} - x_{p^l(k+1)}^l$ is as small as possible. Theoretically, these requirements are supported by the fact that the polynomial reproduction property of low-pass filters (and the vanishing moment property of wavelet filters) can be exploited best for equidistantly sampled functions on a line, see Section 4. Of course, the two conditions (3.3) and (3.4) can also be coupled.

3.2. Adaptive random path construction

Instead of the adaptive path construction considered above, we may also apply a procedure, where the next path component is taken randomly from the remaining indices, and where the probability to choose the next index depends on the spatial distance of points on the one hand and on the distance of the corresponding function values on the other hand.

For the path construction at the l -th level, we now consider the vectors $y_j^l = ((x_j^l)^T, c_j^l)^T \in \mathbb{R}^{d+1}$ (with $c_j^0 := \tilde{f}(x_j^0) = \tilde{f}(x_j)$) and define a symmetric weight matrix $W^l = (w(y_i^l, y_j^l))_{i,j=1}^{N/2^l}$, where the weights in the product

$$w(y_i^l, y_j^l) = w_1(x_i^l, x_j^l) \cdot w_2(c_i^l, c_j^l)$$

may be chosen differently, depending on the range of scattered points x_i^l and of the given (noisy) function values resp. low pass values c_i^l . A possible weight function, also used in the context of bilateral filtering [18], is

$$w(y_i^l, y_j^l) = \exp\left(\frac{-\|x_i^l - x_j^l\|_2^2}{2^{2l/d}\sigma_1}\right) \cdot \exp\left(\frac{-|c_i^l - c_j^l|^2}{2^l\sigma_2}\right) \quad (3.5)$$

where σ_1 and σ_2 need to be chosen appropriately. The normalization by $2^{2l/d}$ in the definition of the weight is due to the fact that each level of the wavelet transform involves a decimation of the number of scattered points by two, and the remaining points have a larger distances of each other. Regarding the low pass values, the range of these values grows by $\sqrt{2}$ due to the filter normalization $\sum_{l \in \mathbb{Z}} \tilde{h}_l = \sqrt{2}$.

While the proposed weight in (3.5) leads to a huge fully occupied weight matrix, we can strongly reduce the numerical effort by cutting the spatial weight at a suitable distance, i.e.

$$w_1(x_i^l, x_j^l) = \begin{cases} \exp(-\|x_i^l - x_j^l\|_2^2 / (2^{2l/d}\sigma_1)) & \|x_i^l - x_j^l\|_2 \leq 2^{l/d}D_1, \\ 0 & \|x_i^l - x_j^l\|_2 > 2^{l/d}D_1, \end{cases}$$

where we have to choose D_1 appropriately to ensure a sufficiently large neighborhood for each point x_i^l . Similarly, the weight $w_2(c_i^l, c_j^l)$ may be just put to zero if the distance $|c_i^l - c_j^l|$ is greater than $2^{l/2}D_2$ with a suitable constant D_2 . We propose now the following algorithm.

Algorithm 3.3. (Random path construction)

Given: the point set $\Gamma^l = \{x_1^l, \dots, x_{N/2^l}^l\} \subset \mathbb{R}^d$, set of vectors $y_j^l = ((x_j^l)^T, c_j^l)^T$, low pass values c_j^l , $j = 1, \dots, N/2^l$, $N = 2^t$.

1. Compute the weight matrix $W^l = (w(y_i^l, y_j^l))_{i,j=1}^{N/2^l}$.
2. Choose the first path component $p^l(1)$ randomly from the index set $\{1, \dots, N/2^l\}$.
3. Considering the row $p^l(1)$ of the weight matrix W^l , let

$$s_{p^l(1)} := \sum_{\substack{j=1 \\ j \neq p^l(1)}}^{N/2^l} w(y_{p^l(1)}^l, y_j^l)$$

be the sum of weights in this row obtained by disregarding the diagonal element $w(y_{p^l(1)}^l, y_{p^l(1)}^l)$, and let now for $r \in \{1, \dots, N/2^l\} \setminus \{p^l(1)\}$,

$$P_{p^l(1),r} := \frac{w(y_{p^l(1)}^l, y_r^l)}{s_{p^l(1)}}$$

be the probability to choose r as the next component $p^l(2)$ in the path vector. Choose now $p^l(2)$ randomly from $\{1, \dots, N/2^l\} \setminus \{p^l(1)\}$ according to the determined probability distribution.

4. Generally, proceed with the following iteration for $k = 1, \dots, N/2^l - 1$. After having fixed the first k components $p^l(1), \dots, p^l(k)$ in the path vector, we consider the submatrix W_{k-1}^l obtained by deleting the $p^l(1)$ -th, \dots , $p^l(k-1)$ -th rows and columns of W^l , and apply the same procedure as before. We determine

$$s_{p^l(k)} = \sum_{\substack{j=1 \\ j \notin \{p^l(1), \dots, p^l(k)\}}}^{N/2^l} w(y_{p^l(k)}^l, y_j^l)$$

and compute the probability

$$P_{p^l(k),r} := \frac{w(y_{p^l(k)}^l, y_r^l)}{s_{p^l(k)}} \quad \text{for } r \in \{1, \dots, N/2^l\} \setminus \{p^l(1), \dots, p^l(k)\}.$$

Choose $p^l(k+1)$ randomly from $\{1, \dots, N/2^l\} \setminus \{p^l(1), \dots, p^l(k)\}$ according to the determined probability distribution.

Output: path vector $p^l = (p^l(k))_{k=1}^{N/2^l}$.

Remark 3.4.

1. For the two proposed path constructions, the choice of path vectors depends on the given (noisy) data values $c_j^0 = \tilde{f}(x_j)$ and the low-pass filtered values c_j^l and hence is adaptive. One may alternatively construct completely non-adaptive path vectors by taking into account only the neighborhoods of the given scattered points x_j^l at each level l (or only the spatial weights $w_1(x_i^l, x_j^l)$) and not the data values c_j^l . A non-adaptive denoising procedure possesses the advantage that all path vectors can be computed beforehand,

such that a wavelet shrinkage procedure is similarly fast as a tensor product wavelet shrinkage in the regular case. However, the results of the non-adaptive denoising approach are worse than the performance using the adaptively determined path vectors, see Section 5.

2. Our proposed adaptive path constructions significantly differ from the generalized tree constructions in [9] that depend only on the set of scattered points x_j and are hence non-adaptive.

3. The idea for determining the distribution for the random path vector construction is slightly related with the ideas for graph Laplacian constructions, see [5].

4. Properties of wavelet transform on paths

Usual wavelet filter banks possess a lot of favorable properties as polynomial reproduction of low-pass filters resp. vanishing moments of wavelet filters. For a given one-dimensional function f with a certain smoothness, a corresponding decay of wavelet coefficients $d_k^l(f)$ in the wavelet expansion

$$f(x) = \sum_{l,k} d_k^l(f) \psi_{l,k}(x)$$

can be shown. Within the last years, this property has been extensively exploited for function space characterizations by means of wavelet expansions. Conversely, wavelet coefficients of high magnitude indicate discontinuities (or points of lower smoothness). The success of wavelet denoising algorithms by thresholding of wavelet coefficients is based on these properties, since for (piecewise) smooth functions, small wavelet coefficients are related to noise. Considering now the case of scattered data $\{x_1, \dots, x_N\}$ in the multivariate case and the wavelet transform along certain path vectors, we may ask, whether the favorable properties of the wavelet transform at least partially transfer to this setting. Regarding the polynomial reproduction of the low-pass filters along path vectors of scattered data, we find

Theorem 4.1. *Let $(\tilde{h}_k)_{k \in \mathbb{Z}}$ be the low-pass filter of the decomposition step in a wavelet filter bank of perfect reconstruction with $\sum_{k \in \mathbb{Z}} \tilde{h}_k = 1$. Further, let $p^1 = (p^1(j))_{j=1}^N$ be the path vector for ordering the scattered data points $x_j = (x_{j,1}, \dots, x_{j,d})^T \in \mathbb{R}^d$. Applying the low-pass filter \tilde{h} to the sequence $(f(x_j))_{j=1}^N$, where $f(x) = a^T x + b$ is a d -variate linear polynomial with $a \in \mathbb{R}^d$, $b \in \mathbb{R}$, the sequence of low-pass coefficients $(c_j^1)_{j=1}^{N/2}$ obtained by one decomposition step of the wavelet filter bank along the path p^1 again reproduces f at the points $(x_j^1)_{j=1}^{N/2}$ with $x_j^1 = \sum_{k \in \mathbb{Z}} \tilde{h}_k x_{p^1(2j-1-k)}$, i.e., we have*

$$f(x_j^1) = c_j^1.$$

Proof. Application of the filter $(\tilde{h}_k)_{k \in \mathbb{Z}}$ to the sequence of function values along the path p^1 gives

$$(c_j^1)_{j=1}^{N/2} := \left(\sum_{k \in \mathbb{Z}} \tilde{h}_k f(x_{p^1(2j-1-k)}) \right)_{j=1}^{N/2}$$

with

$$c_j^1 = \sum_{k \in \mathbb{Z}} \tilde{h}_k (a^T x_{p^1(2j-1-k)} + b) = a^T \left(\sum_{k \in \mathbb{Z}} \tilde{h}_k x_{p^1(2j-1-k)} \right) + b \sum_{k \in \mathbb{Z}} \tilde{h}_k.$$

Hence the assertion of the theorem follows using the new sequence of scattered points x_j^1 with

$$x_j^1 = \sum_{k \in \mathbb{Z}} \tilde{h}_k x_{p^1(2j-1-k)},$$

where $(x_j^1)_{l=1}^{N/2}$ is obtained by (componentwisely) applying the low-pass filter \tilde{h} and downsampling to the sequence of scattered points $(x_{p^1(j)})_{j=1}^N$ ordered along the path p^1 . \square

We suppose that the wavelet filter $\tilde{g} = (\tilde{g}_k)_{k \in \mathbb{Z}}$ in the decomposition step of the wavelet filter bank satisfies the moment conditions $\sum_{k \in \mathbb{Z}} \tilde{g}_k = 0$ and $\sum_{k \in \mathbb{Z}} k \tilde{g}_k = 0$. Considering the wavelet coefficients obtained by applying the wavelet filter \tilde{g} along the path vector p^1 followed by downsampling, we observe that a constant function $f(x) = c$, $c \in \mathbb{R}$, yields the wavelet coefficients

$$d_j^1 = \sum_{k \in \mathbb{Z}} \tilde{g}_k f(x_{p^1(2j-1-k)}) = c \cdot \sum_{k \in \mathbb{Z}} \tilde{g}_k = 0,$$

while for a linear polynomial $f(x) = a^T x + b$ with $a \in \mathbb{R}^d$ and $b \in \mathbb{R}$, the wavelet coefficients

$$d_j^1 = \sum_{k \in \mathbb{Z}} \tilde{g}_k (a^T x_{p^1(2j-1-k)} + b) = a^T \left(\sum_{k \in \mathbb{Z}} \tilde{g}_k x_{p^1(2j-1-k)} \right) + b \sum_{k \in \mathbb{Z}} \tilde{g}_k$$

can only vanish if the sequence $(x_{p^1(2j-1-k)})_{k \in \mathbb{Z}}$ is linear in each component. This observation implies that path vectors p^l should locally be chosen such that the values in the d component sequences $(x_{p^1(2j-1-k), \nu})_{k \in \mathbb{Z}}$, $\nu = 1, \dots, d$, lie approximately equidistantly on a line if f is smooth in this region, see (3.3) and (3.4) in Algorithm 3.1.

Regarding the decay of wavelet coefficients obtained by a wavelet transform along paths (using a different path at each level), we can apply the results of [15] for two-dimensional piecewise Hölder continuous functions of order $\alpha \in (0, 1]$ thereby supporting our procedure of data denoising via wavelet transforms along paths.

Let $\{\Omega_i\}_{1 \leq i \leq K}$ be a finite set of regions in $[0, 1]^2$ such that $\bigcup_{i=1}^K \Omega_i = [0, 1]^2$ and $\Omega_i \cap \Omega_j = \emptyset$ for $i \neq j$, and where $\bar{\Omega}_i$ is assumed to be a connected subset of $[0, 1]^2$ for each $i = 1, \dots, K$. Further, let us assume that the bivariate function f satisfies a Hölder condition in each region Ω_i , $1 \leq i \leq K$, i.e.

$$|f(x) - f(x+h)| \leq C \|h\|_2^\alpha \quad \text{for } x, x+h \in \Omega_i$$

for some $\alpha \in (0, 1]$ and $C > 0$ independent of i . Let the function f be uniformly sampled on Ω , i.e., we have given $f(x_{j_1, j_2})$ with $x_{j_1, j_2} \in I_{2^J} := \{(\frac{j_1}{2^J}, \frac{j_2}{2^J}), j_1, j_2 = 0, \dots, 2^J - 1\}$, and $N = 2^{2J}$. We can perform up to $L = 2J$ levels of the wavelet transform along path vectors (EPWT).

For such “cartoon-like” functions, it has been shown in [15] that the EPWT wavelet coefficients corresponding to one region Ω_i satisfy the estimate

$$|d_k^l(f)| \leq \frac{1}{2} C D^\alpha 2^{-(2J-l)(\alpha+1)/2},$$

where C is the Hölder constant and α the Hölder exponent. The constant D measures the uniformity of the scattered points after each level, where we assume that

$$\min |x_j^l - x_k^l| < D 2^{l/2}.$$

This condition is described as “diameter condition” in [15] and can be interpreted as a condition on the choice of the path vectors. Moreover, it has been shown in [15] that for $J \rightarrow \infty$, the EPWT leads to an asymptotically optimal N -term approximation f_N of f satisfying the estimate

$$\|f - f_N\|_{L^2(\Omega)}^2 \leq \tilde{C} N^{-\alpha},$$

where f_N is the two-dimensional function that is reconstructed from the N most significant EPWT-wavelet coefficients. The above optimal approximation result holds under the assumption that the path vectors p^l at each level of the EPWT transform satisfy (besides the diameter condition) a second condition, termed “region condition” [15]. The region condition ensures that the next index in the path vector p^l should be taken from the same region Ω_i if possible, before crossing to another region. The two conditions on the path vectors required in [15] are both contained in our algorithm. In the adaptive deterministic path construction, the neighborhood definition in (3.1) ensures the diameter condition while the neighborhood definition $N_\theta(x)$ in (3.2) provides the region condition. In the random path construction, the two path conditions are included by the choice of the weight that produces higher probabilities for those indices to be the next path component whose spatial distance $\|x_{p^l(k)}^l - x_{p^l(k+1)}^l\|_2$ as well as distance of function values/low pass values $|c_{p^l(k)}^l - c_{p^l(k+1)}^l|$ are small.

Hence the N -term approximation result in [15] can be taken as a further evidence, that small wavelet coefficients are rather due to noise than to important signal structure.

5. Numerical results

We want to apply the proposed scattered data wavelet denoising algorithm along path vectors to 256×256 pixel images. The noisy images in this subsection are generated by adding synthetic white Gaussian noise to the original

natural images. The quality of the denoising results is measured by the peak signal to noise ratio (PSNR) given by

$$\text{PSNR} = 10 \log_{10} \frac{\max_{j=1, \dots, N} f(x_j)}{\frac{1}{N} \sum_{j=1}^N (\tilde{f}(x_j) - f(x_j))^2},$$

where f denotes the original and \tilde{f} the noisy image. In our experiments we have taken $\max_{j=1, \dots, N} f(x_j) = 255$.

In Figure 1, we present the original pepper image (a), the noisy pepper image (b) with PSNR = 19.97, and the denoising results using several different algorithms. In particular, we consider the 7-9 biorthogonal tensor-product wavelet shrinkage (c) with a PSNR of 24.91, the 7-9 biorthogonal wavelet shrinkage with cycle spinning (d) with 64 shifts and a PSNR of 28.11, the four-pixel scheme by Welk, Steidl and Weickert [20] (e) using 76 iterations and a step size $\tau = 0.001$ providing a PSNR of 28.26, and the curvelet shrinkage (e) with best threshold parameter 80 yielding a PSNR of 26.36. The curvelet denoising result is obtained with the help of the CurveLab software that is available at www.curvelet.org, see also [4]. The results of our algorithm (using a 7-9 biorthogonal wavelet filter bank) with a deterministic path vector (g) and a random path vector (h) are given in Figures 1(g) and 1(h). For the tensor-product wavelet shrinkage we have used the shrinkage parameter $\theta = 74$. For the deterministic path vector construction in (3.2) we have taken $C_1 = 1.3$ and shrinkage parameter $\theta = 89$. For the random path vectors that perform slightly worse than the deterministic vectors, we have taken the weight with $\sigma_1 = \sigma_2 = 0.2$, $D_1 = 5.0$ and $\theta = 71$. Further, we have 64 times applied Algorithm 2.1 and have averaged the results. The parameters for the curvelet transform and for the four-pixel scheme have been taken such that they perform optimally.

In Figure 2, we present the denoising results for the noisy pepper image with a PSNR of 16.45. Again, we compare our denoising results with tensor-product shrinkage, the four-pixel scheme and with curvelet shrinkage. For the tensor product case with the biorthogonal 7-9 filter bank we have used the optimal threshold $\theta = 140$ (without cycle spinning) and $\theta = 115$ (with cycle spinning). The four-pixel scheme works best with a time step of 0.001 and 124 iterations. The curvelet shrinkage uses the optimal thresholds $\theta = 106$. For our method with the deterministic path we have taken the threshold $\theta = 127$ and $C_1 = 1.1$; for the random path the same threshold and a weight with $\sigma_1 = \sigma_2 = 0.2$ and with $D_1 = 5$.

The denoising results for the pepper image and for the cameraman image of size 256×256 are summarized in Table 1, where we have chosen optimal parameters for each method.

In Figure 3, we illustrate the remaining scattered points obtained after different levels of our deterministic path algorithm that is applied to the cameraman image. The results show that the points are rather well distributed. These points are at a time used for constructing the next path vector.

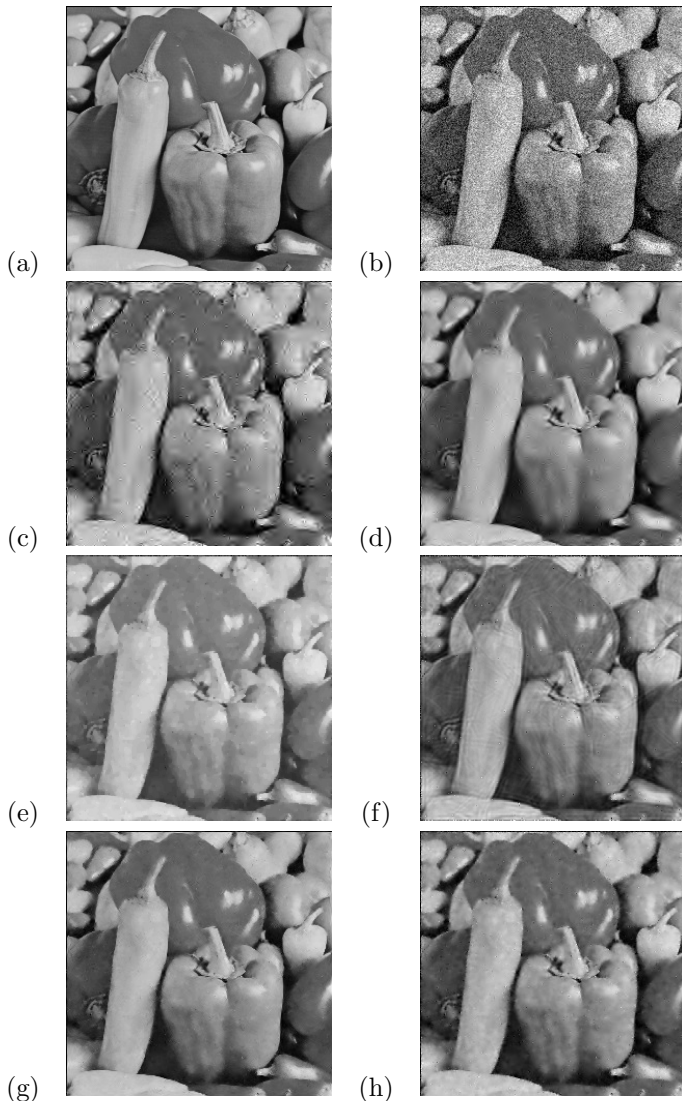


Figure 1. Top: (a) original image, (b) noisy image, PSNR 19.97; second row: denoising by tensor product biorthogonal 7-9 wavelet transform (c) without cycle spinning, PSNR 24.91; (d) with cycle spinning with 64 shifts, PSNR 28.11; third row: four pixel scheme, PSNR 28.26; (f) curvelet shrinkage, PSNR 26.36; last row: our scheme with (g) deterministic path, PSNR 29.01; (h) random path, PSNR 27.96.

Finally, we will show a denoising experiment, where we no longer consider a rectangular region but a region with an L -form, see Figure 4. Here we have used deterministic path vectors with threshold $\theta = 89$ and with $C_1 = 1.3$. The main advantage of our new scattered data denoising algorithm

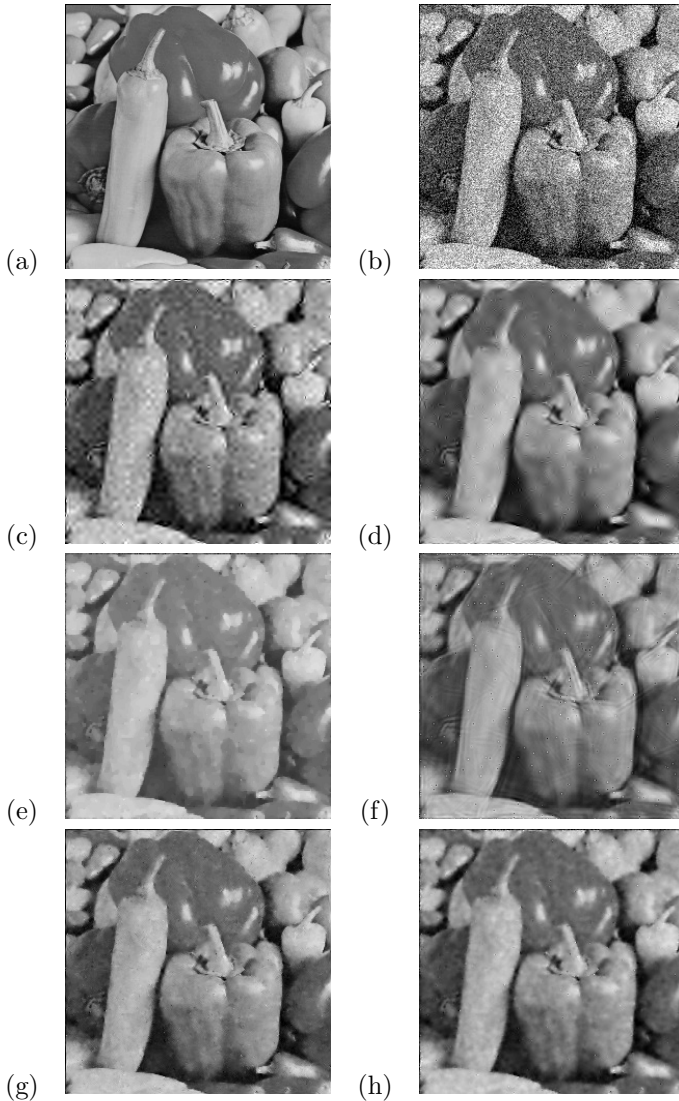


Figure 2. Top: (a) original image, (b) noisy image, PSNR 16.45; second row: denoising by tensor product biorthogonal 7-9 wavelet transform (c) without cycle spinning, PSNR 23.20; (d) with cycle spinning with 64 shifts, PSNR 25.86; third row: four pixel scheme, PSNR 26.13; (f) curvelet shrinkage, PSNR 23.95; last row: our scheme with (g) deterministic path, PSNR 26.44; (h) random path, PSNR 25.69.

is that we can apply it to denoise functions on any connected region, where we have to adjust step 3 of Algorithm 2.1 suitably as e.g. given in Remark 2.2.

| image | noisy image | tensor 7-9 | cycle 7-9 | [20] | curvelet shrink. | determ. path | random path |
|-----------|-------------|------------|-----------|-------|------------------|--------------|-------------|
| pepper | 19.97 | 24.91 | 28.11 | 28.26 | 26.36 | 29.01 | 27.96 |
| pepper | 16.45 | 23.20 | 25.86 | 26.13 | 23.95 | 26.44 | 25.69 |
| cameraman | 19.97 | 24.74 | 27.19 | 27.64 | 25.48 | 28.28 | 27.44 |
| cameraman | 16.45 | 22.86 | 25.14 | 25.73 | 23.73 | 26.15 | 24.85 |

Table 1. Comparison of our scattered data denoising method with deterministic and random path vectors with other denoising methods, see also Figures 1 and 2.

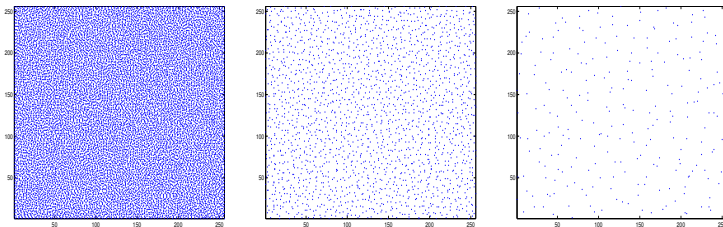


Figure 3. Scattered points after the second level (left), the fifth level (middle) and the eighth level (right) of the denoising algorithm.



Figure 4. L-sector of the cameraman image, original (left), noisy image with PSNR of 19.97 (middle) and denoised image with the proposed method using a deterministic path vector with a PSNR of 27.77.

Acknowledgment

This work is supported by the priority program SPP 1324 of the Deutsche Forschungsgemeinschaft (DFG), project PL 170/13-2.

References

- [1] A. Antoniadis and D.T. Pham, Wavelet regression for random or irregular design, *Comput. Statist. Data Anal.* **28** (1998), 353–369.
- [2] R. Baraniuk, A. Cohen, and R. Wagner, Approximation and compression of scattered data by meshless multiscale decompositions, *Appl. Comput. Harmon. Anal.* **25** (2008), 133–147.
- [3] T. Cai and L.D. Brown, Wavelet shrinkage for nonequispaced samples, *Ann. Statist.* **26**(5) (1998), 1783–1799.

- [4] E. Candes, L. Demanet, D. Donoho, and L. Ying, Fast discrete curvelet transforms, *Multiscale Model. Simul.* **5** (2006), 861–899.
- [5] F. Chung, Spectral Graph Theory, CBMS Regional Conference Series in Mathematics **92**, Amer. Math. Soc., Providence, RI, 1997.
- [6] R.R. Coifman and D.L. Donoho, Translation-invariant de-noising, in *Wavelets and Statistics*, A. Antoniadis and G. Oppenheim, eds., Springer-Verlag, New York, 1995, pp. 125–150.
- [7] V. Delouille, Nonparametric stochastic regression using design-adapted wavelets, Ph.D. Thesis, Universite Catholique de Louvain, 2002.
- [8] H.-Y. Gao and A. G. Bruce, WaveShrink with firm shrinkage, *Statistica Sinica* **7** (1997), 855–874.
- [9] M. Gavish, B. Nadler, and R.R. Coifman, Multiscale wavelets on trees, graphs and high-dimensional data: Theory and applications to semi supervised learning, *Proceedings of the 27th International Conference on Machine Learning*, 2010.
- [10] P. Hall and B.A. Turlach, Interpolation methods for nonlinear wavelet regression with irregularly spaced design. *Ann. Statist.* **25**(5) (1997), 1912–1925.
- [11] M. Jansen, G. Nason, and B. Silverman, Scattered data smoothing by empirical Bayesian shrinkage of second generation wavelet coefficients, Wavelet App. in Sig. and Image Processing IX, Proc. of SPIE. 4478 (2001), 87–97.
- [12] A. Kovac and B.W. Silverman, Extending the scope of wavelet regression methods by coefficient-dependent thresholding. *J. A. Statist. Assoc.* **95** (2000), 172–183.
- [13] G. Plonka, The easy path wavelet transform: a new adaptive wavelet transform for sparse representation of two-dimensional data, *Multiscale Model. Simul.* **7** (2009), 1474–1496.
- [14] G. Plonka and D. Roşca, Easy path wavelet transform on triangulations of the sphere, *Mathematical Geosciences* **42**(7) (2010), 839–855.
- [15] G. Plonka, S. Tenorth, and A. Iske, Optimally Sparse Image Representation by the Easy Path Wavelet Transform. *Int. J. Wavelets Multiresolut. Inf. Process.* (2012), to appear.
- [16] I. Ram, M. Elad, and I. Cohen, Generalized tree-based wavelet transform, IEEE Trans. Signal Process. **59**(9) (2011), 4199–4209.
- [17] W. Sweldens, The lifting scheme: A construction of second generation wavelets, *SIAM J. Math. Anal.* **29**(2) (1997), 511–546.
- [18] C. Tomasi and R. Manduchi, Bilateral filtering for gray and color images, in *Proc. 6th Int. Conf. Computer Vision*, New Delhi, India, 1998, 839–846.
- [19] E. Vanraes, M. Jansen, and A. Butheel, Stabilized wavelet transforms for non-equispaced data smoothing. *Signal Process.* **82**(12) (2002), 1979–1990.
- [20] M. Welk, J. Weickert, and G. Steidl, A four-pixel scheme for singular differential equations. In: R. Kimmel, N. Sochen, J. Weickert (Eds.), *Scale-Space and PDE Methods in Computer Vision*, Lecture Notes in Computer Science 3459, Springer, Berlin (2005) 610–621.

Dennis Heinen
University of Göttingen
Institute for Numerical and Applied Mathematics
Lotzestr. 16-18
37083 Göttingen
Germany
e-mail: d.heinen@math.uni-goettingen.de

Gerlind Plonka
University of Göttingen
Institute for Numerical and Applied Mathematics
Lotzestr. 16-18
37083 Göttingen
Germany
e-mail: plonka@math.uni-goettingen.de