Adaptive Approximation Algorithms for Sparse Data Representation

M. Guillemard, D. Heinen, A. Iske, S. Krause-Solberg, and G. Plonka

Abstract We survey our latest results on the development and analysis of adaptive approximation algorithms for sparse data representation, where special emphasis is placed on the Easy Path Wavelet Transform (EPWT), nonlinear dimensionality reduction (NDR) methods, and their application to signal separation and detection.

1 Introduction

During the last few years there has been an increasing interest in efficient (i.e., sparse) representation and denoising of high-dimensional signals. We have focussed our research on the development and analysis of adaptive approximation algorithms for high-dimensional signals, especially (a) scattered data denoising by wavelet transforms; (b) nonlinear dimensionality reduction relying on geometrical and topological concepts. This contribution reviews our recent research results on (a) and (b).

For (a), we present a general framework for the *Easy Path Wavelet Transform* (EPWT) for sparse representation and denoising of scattered data taken from high-dimensional signals (in Section 2). As regards (b), we continue our research on nonlinear dimensionality reduction (NDR) methods (cf. Section 3), where we combine recent NDR methods with non-negative matrix factorization (NNMF), for the purpose of separating sources from a mixture of signals without a prior knowledge about the mixing process. More details on dimensionality reduction and NNMF, along with our recent results on signal separation, are discussed in Section 4.

Mijail Guillemard
Technische Universität Berlin, 10623 Berlin. e-mail: guillemard@math.tu-berlin.de

Dennis Heinen, Gerlind Plonka
Universität Göttingen, 37083 Göttingen. e-mail: {d.heinen,plonka}@math.uni-goettingen.de

Armin Iske, Sara Krause-Solberg
Universität Hamburg, 20146 Hamburg. e-mail: {armin.iske,sara.krause-solberg}@uni-hamburg.de

The presented results are based on our papers [7-8,10,12,16-20,24] and have been achieved in the project "Adaptive approximation algorithms for sparse data representation" of the German Research Foundation's priority program DFG-SPP 1324.

2 The Easy Path Wavelet Transform

Let Ω be a connected domain in $\mathbb{R}^d$ and let Γ be a large finite set of points in Ω. We let $h_\Gamma := \max_{y\in\Omega} \min_{x\in\Gamma} \|y-x\|_2$ be the *fill distance* of Γ in Ω and its *grid distance* is $g_\Gamma := \min_{x,x'\in\Gamma, x\neq x'} \|x-x'\|_2$. We say that the set Γ is *quasi-uniform*, if $h_\Gamma < 2g_\Gamma$. Further, let $f : \Omega \to \mathbb{R}$ be a piecewise smooth function that is sampled at Γ, i.e., the values $f(x)$, $x \in \Gamma$, are given. We are now interested in an efficient approximation of f using a meshless multiscale approach called Easy Path Wavelet Transform (EPWT). For applications, we usually assume that Γ approximates a smooth manifold in $\mathbb{R}^d$. For example, our approach covers the efficient approximation of digital images, see [17, 21], where Γ is chosen to be a set of regular grid points in a rectangle Ω, and the approximation of piecewise smooth functions on the sphere, see [19], where $\Omega = \mathbb{S}^2$ and Γ is a suitably chosen quasi-uniform point set on the sphere $\mathbb{S}^2$.

Similar approaches have also been proposed for generalizing the wavelet transform to data defined on weighted graphs, see [23]. In this section, we extend the EPWT proposed in [17, 19, 25] to the case of high-dimensional data approximation.

2.1 The General EPWT Algorithm for Sparse Approximation

Let us shortly recall the notions of a biorthogonal wavelet filter bank of perfect reconstruction. To this end, let φ be a sufficiently smooth, compactly supported, one-dimensional scaling function, $\tilde{\varphi}$ the corresponding biorthogonal compactly supported scaling function, and ψ, $\tilde{\psi}$ the corresponding pair of biorthogonal compactly supported wavelets, see, e.g., [3, 16]. These functions provide us with a filter bank of perfect reconstruction with sequences $(h_n)_{n\in\mathbb{Z}}$, $(\tilde{h}_n)_{n\in\mathbb{Z}}$ of low-pass filter coefficients and $(g_n)_{n\in\mathbb{Z}}$, $(\tilde{g}_n)_{n\in\mathbb{Z}}$ of high-pass filter coefficients.

Assume that the number $N = |\Gamma|$ of given points $x \in \Gamma \subset \mathbb{R}^d$ is a power of 2, $N = 2^J$, where $J \gg 1$. We denote $\Gamma^J := \Gamma$ and its elements by $x_k^J = x_k$, $k = 1,\dots,N$, i.e., we fix some ordering of the points in Γ^J. Now the EPWT works as follows. In a first step, we seek a suitable permutation p^J of the indices of the points in Γ^J by determining a path of length N through all points x_k^J such that consecutive data points $(x^J_{p^J(k)}, f(x^J_{p^J(k)}))$ and $(x^J_{p^J(k+1)}, f(x^J_{p^J(k+1)}))$ in the path strongly "correlate". In the second step, we apply the one-dimensional wavelet filter bank to the sequence of functions values $(f(x^J_{p^J(k)}))_{k=1}^N$, and simultaneously a low-pass filter to the points $(x^J_{p^J(k)})_{k=1}^N$, where we consider each of the d components separately. The significant

high-pass coefficients corresponding to the function values will be stored. The $N/2$ low-pass data will be processed further at the next level of the EPWT. Particularly, we denote the set of the $N/2$ points obtained by low-pass filtering and downsampling of $(x^J_{p^J(k)})_{k=1}^N$ by Γ^{J-1}, and relate the low-pass function coefficients to these points. Again, we start with seeking a permutation p^{J-1} of the indices of the points in Γ^{J-1} to obtain an appropriate ordering of the data and apply the one-dimensional wavelet filter bank to the ordered low-pass function data. We iterate this procedure and obtain a sparse representation of the original data by applying a hard thresholding procedure to the high-pass coefficients of the function value components. The complete procedure can be summarized as follows.

Algorithm 1 (Decomposition) *Let $\Gamma = \{x_1,\dots,x_N\} = \{x^J_1,\dots,x^J_N\} = \Gamma^J \subset \mathbb{R}^d$ be a given point set. Let $f^J_k := f(x_k)$, for $k = 1,\dots,N$, where $N = 2^J$. Choose a biorthogonal wavelet filterbank with decomposition filters $\tilde{h},\tilde{g}$, and reconstruction filters h,g, where $\sum_{k\in\mathbb{Z}} \tilde{h}(k) = \sqrt{2}$, and a low-pass filter $\tilde{h}_p$, where $\sum_{k\in\mathbb{Z}} \tilde{h}_p(k) = 1$.*

Iteration: Perform the following 4 steps for $\ell = J, J-1\dots, J-L+1$ with $L < J$:

1. *Find a suitable path vector $p^\ell \in \mathbb{N}^{2^\ell}$ consisting of a permutation of the indices of the points in Γ^ℓ that describes a fixed order of points $(x^\ell_{p^\ell(k)}, f^\ell_{p^\ell(k)})$, $k = 1,\dots,2^\ell$.*
2. *Apply the (periodic) low-pass filter $\tilde{h}$ to $(f^\ell_{p^\ell(k)})_{k=1}^{2^\ell}$ followed by downsampling by two to obtain the low-pass data $(f^{\ell-1}_k)_{k=1}^{2^{\ell-1}}$. Apply the (periodic) high-pass filter $\tilde{g}$ to $(f^\ell_{p^\ell(k)})_{k=1}^{2^\ell}$ followed by downsampling by two to obtain the vector of wavelet coefficients $(d^{\ell-1}_k)_{k=1}^{2^{\ell-1}}$.*
3. *Apply the low-pass filter $\tilde{h}_p$ to point vector $(x^\ell_{p^\ell(k)})_{k=1}^{2^\ell}$ (component-wise) followed by downsampling by two to obtain a new vector of scattered points $(x^{\ell-1}_k)_{k=1}^{2^{\ell-1}}$. Determine the new point set $\Gamma^{\ell-1} := \{x^{\ell-1}_1,\dots,x^{\ell-1}_{2^{\ell-1}}\}$.*
4. *Apply a hard-threshold operator T_θ to the wavelet vector $(d^{\ell-1}_k)_{k=1}^{2^{\ell-1}}$ to find*

$$\tilde{d}^{\ell-1}_k = T_\theta(d^{\ell-1}_k) = \begin{cases} d^{\ell-1}_k & \text{if } |d^{\ell-1}_k| \geq \theta, \\ 0 & \text{if } |d^{\ell-1}_k| < \theta, \end{cases}$$

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

Output: low-pass function values $(f^{J-L}_k)_{k=1}^{2^{J-L}}$, thresholded high-pass function values $(\tilde{d}^\ell_k)_{k=1}^{2^\ell}$, $\ell = J-1,\dots,J-L$, path vectors p^ℓ, $\ell = J,\dots,J-L+1$.

By construction many high pass values d^ℓ_k will vanish. An optimal storage of the path vectors p^ℓ depends on the original distribution of the points x^J_k and on the applied filter $\tilde{h}_p$. Employing a "lazy" filter, we have $x^\ell_k := x^{\ell+1}_{p^{\ell+1}(2k)}$, such that at each level the new point set is just a subset of that of the preceding level of half cardinality.

Algorithm 2 (Reconstruction) *Reconstruct values $f(x_k) = f(x_k^J)$ by applying the following iteration, where $(\tilde{f}_k^{J-L})_{k=1}^{2^{J-L}} := (f_k^{J-L})_{k=1}^{2^{J-L}}$.*

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

1. *Apply an upsampling by two and then the low-pass filter h to $(\tilde{f}_k^{\ell})_{k=1}^{2^{\ell}}$.*
2. *Apply an upsampling by two and then the high-pass filter g to $(\tilde{d}_k^{\ell})_{k=1}^{2^{\ell}}$.*
3. *Add the results of the previous two steps to obtain $(\tilde{f}^{\ell+1}_{p^{\ell+1}(k)})_{k=1}^{2^{\ell+1}}$, and invert permutation $p^{\ell+1}$.*

Output: $(\tilde{f}_k^J)_{k=1}^N$, the approximated function values at scattered points $x_k \in \Gamma$.

2.2 Construction of Path Vectors

The main challenge for the application of the EPWT to sparse data representation is to construct path vectors through the point sets Γ^{ℓ}, $\ell = J, \dots, J-L+1$. This step is crucial for the performance of the data compression. The path construction is based on determining a suitable correlation measure that takes the local distance of the scattered points x_k^{ℓ} into account, on the one hand, and the difference of the corresponding low-pass values f_k^{ℓ}, on the other hand. In the following, we present some strategies for path construction and comment on their advantages and drawbacks.

2.2.1 Path Construction with Fixed Local Distances

One suitable strategy for path construction [17, 25] is based on a priori fixed local ε-neighborhoods of the points x_k^{ℓ}. In $\mathbb{R}^d$, we consider a neighborhood of the form

$$N_{\varepsilon}(x_k^{\ell}) = \{x \in \Gamma^{\ell} \setminus \{x_k^{\ell}\} : \|x_k^{\ell} - x\|_2 \le 2^{-\ell/d}\varepsilon\},$$

where $\varepsilon > 2^{J/d} g_{\Gamma}$ depends on the distribution of the original point set $\Gamma = \Gamma^J$. For example, starting with a regular rectangular grid in $\mathbb{R}^2$ with mesh size $g_{\Gamma} = 2^{-J/2}$ (with J even) in both directions, one may think about a constant ε with $\sqrt{2} \le \varepsilon < 2$, such that each inner grid point has 8 neighbors.

For path construction at level ℓ of the EPWT, we choose a first point $x^{\ell} \in \Gamma^{\ell}$ randomly, and put $x^{\ell}_{p^{\ell}(1)} := x^{\ell}$. Let now $P_j^{\ell} := \{x^{\ell}_{p^{\ell}(1)}, \dots, x^{\ell}_{p^{\ell}(j)}\}$ be the set of points that have already been taken in the path. Now, we determine the $(j+1)$-th point by

$$x^{\ell}_{p^{\ell}(j+1)} := \operatorname*{argmin}_{x \in N_{\varepsilon}(x^{\ell}_{p^{\ell}(j)}) \setminus P_j^{\ell}} |f(x) - f(x^{\ell}_{p^{\ell}(j)})|, \tag{1}$$

i.e., we choose the point x in the neighborhood of the point $x^{\ell}_{p^{\ell}(j)}$ with minimal absolute difference of the corresponding function values. This measure has been applied

in the *rigorous EPWT* of [17, 19]. The advantage of fixing the local neighborhood in spatial domain lies in the reduced storage costs for the path vector that needs to be kept to ensure a reconstruction. The drawback of this measure is that the set of "admissible points" $N_\varepsilon(x^\ell_{p^\ell(j)}) \setminus P^\ell_j$ may be empty. In this case a different rule for finding the next path entry has to be applied.

A special measure occurs if one tries to mimic the one-dimensional wavelet transform. In order to exploit the piecewise smoothness of the function f to be approximated, one should prefer to construct path vectors, where locally three consecutive points $x^\ell_{p^\ell(j-1)}, x^\ell_{p^\ell(j)}, x^\ell_{p^\ell(j+1)}$ lie (almost) on a straight line. This consideration leads to the following measure: We fix a threshold μ for the function values. For finding the next point in the path, we compute

$$N_{\varepsilon,\mu}(x^\ell_{p^\ell(j)}) := \{x \in N_\varepsilon(x^\ell_{p^\ell(j)}) \setminus P^\ell_j : |f(x) - f(x^\ell_{p^\ell(j)})| \le \mu\}, \tag{2}$$

and then let

$$x^\ell_{p^\ell(j+1)} := \operatorname*{argmin}_{x \in N_{\varepsilon,\mu}(x^\ell_{p^\ell(j)})} \frac{\langle x^\ell_{p^\ell(j-1)} - x^\ell_{p^\ell(j)}, x^\ell_{p^\ell(j)} - x\rangle}{\|x^\ell_{p^\ell(j-1)} - x^\ell_{p^\ell(j)}\|_2 \, \|x^\ell_{p^\ell(j)} - x\|_2}, \tag{3}$$

where $\langle\cdot,\cdot\rangle$ denotes the usual scalar product in $\mathbb{R}^d$. Note that in (3) the cosine of the angle between the vectors $x^\ell_{p^\ell(j-1)} - x^\ell_{p^\ell(j)}$ and $x^\ell_{p^\ell(j)} - x$ is minimized if $x^\ell_{p^\ell(j-1)}, x^\ell_{p^\ell(j)}$ and x are co-linear. This approach is taken in [17, 25] for images (called *relaxed EPWT*), and in [11] for scattered data denoising.

Remark 1. The idea to prefer path vectors, where the angles between three consecutive points in the path is as large as possible, can be theoretically validated in different ways. Assume that the given wavelet decomposition filter $\tilde{g} = (\tilde{g}_k)_{k\in\mathbb{Z}}$ in the filter bank satisfies the moment conditions $\sum_{k\in\mathbb{Z}} \tilde{g}_k = 0$ and $\sum_{k\in\mathbb{Z}} k\tilde{g}_k = 0$. Then we simply observe that for a constant function $f(x) = c$ for $x \in \Gamma$ and $c \in \mathbb{R}$ by

$$d^J_n = \sum_{k\in\mathbb{Z}} \tilde{g}_{k-2n} f(x^J_{p^J(k)}) = c \sum_{k\in\mathbb{Z}} \tilde{g}_{k-2n} = 0$$

all wavelet coefficients vanish, while for a linear function of the form $f(x) = a^T x + b$ with $a \in \mathbb{R}^d$ and $b \in \mathbb{R}$ we have

$$d^J_n = \sum_{k\in\mathbb{Z}} \tilde{g}_{k-2n} f(x^J_{p^J(k)}) = a^T \sum_{k\in\mathbb{Z}} \tilde{g}_{k-2n} x^J_{p^J(k)} + b \sum_{k\in\mathbb{Z}} \tilde{g}_{k-2n}.$$

Consequently, these coefficients only vanish, if the points in the sequence $(x^J_{p^J(k)})_{k\in\mathbb{Z}}$ are co-linear and equidistant, see [11]. A second validation for choosing the path vector using the criterion (3) is given by the so-called *path smoothness condition* in [18], see also Subsection 2.4, Remark 4.

Remark 2. Our numerical results in Subsection 2.5 show that the relaxed path construction proposed in (2)-(3) is far superior to the rigorous path construction (1),

since it produces fewer "interruptions", i.e., cases where $N_\varepsilon(x_{p^\ell(j)}) \setminus P_j^\ell = \emptyset$, and a new path entry needs to be taken that is no longer locally correlated to the preceding point, which is usually leading to large wavelet coefficients and a higher effort in path coding (see [17, 25]).

2.2.2 Path Construction with Global Distances

We want to present a second path construction using a global weight function. Considering the vectors $y_k^\ell = y(x_k^\ell) := ((x_k^\ell)^T, f_k^\ell)^T \in \mathbb{R}^{d+1}$ at each level, we define a symmetric weight matrix $W^\ell = (w(y_k^\ell, y_{k'}^\ell))_{k,k'=1}^{2^\ell}$, where the weight is written as

$$w(y_k^\ell, y_{k'}^\ell) = w_1(x_k^\ell, x_{k'}^\ell) \cdot w_2(f_k^\ell, f_{k'}^\ell).$$

Now the weights for the scattered points x_k^ℓ can be chosen differently from the weights for the (low-pass) function values f_k^ℓ. A possible weight function used already in the context of bilateral filtering [26] is

$$w(y_k^\ell, y_{k'}^\ell) = \exp\left(\frac{-\|x_k^\ell - x_{k'}^\ell\|_2^2}{2^{2(J-\ell)/d}\eta_1}\right) \cdot \exp\left(\frac{-|f_k^\ell - f_{k'}^\ell|^2}{2^{J-\ell}\eta_2}\right),$$

where η_1 and η_2 need to be chosen appropriately. The normalization constant $2^{2(J-\ell)/d}$ in the weight w_1 is due to the reduction of the points $x \in \Gamma^\ell$ by factor 2, at each level, so that the distances between the points grow. The normalization constant $2^{J-\ell}$ in the weight w_2 arises from the usual amplification of the low-pass coefficients in the wavelet transform with filters $\tilde{h}$ satisfying $\sum_{k\in\mathbb{Z}} \tilde{h}_k = \sqrt{2}$.

Having computed the weight matrix $W^\ell = (w(y_k^\ell, y_{k'}^\ell))_{k,k'=1}^{2^\ell}$, we simply compute the path vector as follows. We choose the first component $x_{p^\ell(1)}^\ell$ randomly from Γ^ℓ. Using again the notation $P_j^\ell := \{x_{p^\ell(1)}^\ell, \ldots, x_{p^\ell(j)}^\ell\}$ for the set of points in Γ^ℓ that are already contained in the path vector, we now determine the next point as

$$x_{p^\ell(j+1)}^\ell := \operatorname*{argmax}_{x \in \Gamma^\ell \setminus P_j^\ell} w(y(x), y(x_{p^\ell(j)}^\ell)),$$

where uniqueness can be achieved by fixing a rule if the maximum is attained at more than one point. The advantage of this path construction is that no "interruptions" occur. The essential drawback consists in higher storage costs for path vectors, where we can no longer rely on direct local neighborhood properties of consecutive points in the path vector. Further, computing the full weight matrix W^ℓ is very expensive. The costs can be reduced by cutting the spatial weight at a suitable distance defining

$$w_1(x_k^\ell, x_{k'}^\ell) = \begin{cases} \exp(-\|x_k^\ell - x_{k'}^\ell\|_2^2 / 2^{2(J-\ell)/d}\eta_1) & \text{for } \|x_k^\ell - x_{k'}^\ell\|_2 \le 2^{-\ell/d} D, \\ 0 & \text{for } \|x_k^\ell - x_{k'}^\ell\|_2 > 2^{-\ell/d} D, \end{cases} \quad (4)$$

with D chosen appropriately to ensure a sufficiently large spatial neighborhood.

Remark 3. This approach has been used in [11] for random path construction, where the compactly supported weight function $w_1(x_k^\ell, x_{k'}^\ell)$ above is employed. Taking the weight function

$$w_1(x_k^\ell, x_{k'}^\ell) = \begin{cases} 1 & \text{for } \|x_k^\ell - x_{k'}^\ell\|_2 \le 2^{-\ell/d} D, \\ 0 & \text{for } \|x_k^\ell - x_{k'}^\ell\|_2 > 2^{-\ell/d} D, \end{cases}$$

and $w_2(f_k^\ell, f_{k'}^\ell) = \exp\left(\frac{-|f_k^\ell - f_{k'}^\ell|^2}{2^{J-\ell}\eta_2}\right)$ we obtain a distance measure that is equivalent to (1).

2.3 EPWT for Scattered Data Denoising

The EPWT can also be used for denoising of scattered data. Let us again assume $\Gamma = \{x_1, \ldots, x_N\}$ are scattered points in $\mathbb{R}^d$ and let $f : \mathbb{R}^d \to \mathbb{R}$ be a smooth function sampled on $\Gamma \subset \Omega$. For the measured data $\tilde{f}(x_j)$, we suppose that

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

where z_j denotes additive Gaussian noise with zero mean and an unknown variance σ^2. For the distribution of the points in Ω we assume quasi-uniformity as before.

We now apply the EPWT, Algorithms 1 and 2 in Section 2.1, for data denoising. Note that in case of noisy function values, the construction of path vectors (being based on the correlation of function values at points with small spatial distance) is now influenced by the noise. To improve the denoising performance, we have to resemble the "cycle spinning" method (see [4]) that works as follows. We apply the (tensor product) wavelet shrinkage not only to the image itself, but also to the images that are obtained by up to seven cyclic shifts in x- and y-direction. After un-shifting, one takes the average of the 64 reconstructed images, thereby greatly improving the denoising result.

Employing the EPWT algorithm, we use Algorithms 1 and 2, applying them 64 times using different starting values $x_{p^J(1)}$ as a first path component each time. For the path construction, we utilize one of the two methods described in Section 2.2. After reconstruction of the 64 data sets, we take the average in order to obtain the denoising result. Similarly as for wavelet denoising, the threshold parameter θ in Algorithm 1 needs to be selected carefully depending on the noise level.

In [11] we have employed two different path constructions for image denoising. The first one is very similar to the path construction in Subsection 2.2.1. The second one is based on a weight matrix resembling that in Subsection 2.2.2. Here, the next component in the path vector is chosen randomly according to a probability distribution based on the weight matrix.

For images, the proposed denoising procedure strongly outperforms the usual tensor-product wavelet shrinkage with cycle spinning, see [11]. Moreover, the procedure is not restricted to rectangular grids, but can be used in a much more general context for denoising of functions on manifolds. Numerical examples of the EPWT-based denoising scheme are given in Subsection 2.5.

2.4 Optimal Image Representation by the EPWT

In this subsection we restrict ourselves to the EPWT on digital images on a domain $\Omega = [0,1)^2$. For cartoon models, where the image is piecewise Hölder continuous or even Hölder smooth, we can prove that the EPWT leads to optimally sparse image representations, see [18, 20]. To explain this, let $F \in L^2(\Omega)$ be a piecewise Hölder continuous image. More precisely, let $\{\Omega_i\}_{1\le i\le K}$ be a finite set of regions forming a disjoint partition of Ω whose boundaries are continuous and of finite length. In each region Ω_i, F is assumed to be Hölder continous of order $\alpha \in (0,1]$,

$$|F(x)-F(x+h)| \le C\|h\|_2^{\alpha}, \qquad x, x+h \in \Omega_i, \tag{5}$$

where $C > 0$ does not depend on i. For given samples $\{(F(2^{-J/2}n))\}_{n\in I_J}$, the function F can be approximated by the piecewise constant function

$$F^J(x) = \sum_{n\in I_J} F(2^{-J/2}n)\chi_{[0,1)^2}(2^{J/2}x-n), \qquad x \in [0,1)^2,$$

where the index set $I_J := \{n = (n_1,n_2) \in \mathbb{N}^2 : 0 \le n_1 \le 2^{J/2}-1, 0 \le n_2 \le 2^{J/2}-1\}$ is of cardinality 2^J. In this special case $\alpha \in (0,1]$ we can rely on the orthogonal Haar wavelet filter bank in Algorithms 1 and 2. An optimal image representation is strongly based on an appropriate path construction. As shown in [20], we need to satisfy the following two conditions.

Region condition. At each level ℓ of the EPWT, we need to choose the path vector, such that it contains at most $R_1 K$ discontinuities which are incurred by crossing over from one region Ω_i to another region, or by jumping within one region Ω_i. Here R_1 does not depend on J or ℓ, and K is the number of regions.

Diameter condition. At each level ℓ of the EPWT, we require

$$\|x_{p^\ell(k)} - x_{p^\ell(k+1)}\|_2 \le D_1 2^{-\ell/2},$$

for almost all points $x^\ell_{p^\ell(k)}$, $k = 1,\dots,2^\ell-1$, where D_1 does not depend on J or ℓ. The number of path components which do not satisfy the diameter condition is bounded by a constant being independent of ℓ and J.

The region condition suggests that for path construction, we should first collect all points that belong to one region Ω_i before transferring to the next region. The diameter condition ensures that the remaining points in Γ^ℓ are quasi-uniformly dis-

tributed at each level ℓ of the EPWT. Satisfying these two conditions for the path vectors, we have shown in [20], Corollary 3.1 that the M-term approximation F_M reconstructed from the M most significant EPWT wavelet coefficients, satisfies the *optimal* error estimate

$$\|F - F_M\|_2^2 \leq \tilde{C} M^{-\alpha} \tag{6}$$

with a constant $\tilde{C}$ and the Hölder exponent $\alpha \in (0,1]$ in (5).

Remark 4. Observe that at each level of the EPWT the path vector $(p^\ell(j))_{j=1}^{2^j}$ determines a planar curve that interpolates $f^\ell_{p^\ell(j)}$ at the points $x^\ell_{p^\ell(j)}$, $j = 1,\ldots,2^\ell$. By definition, this curve is only piecewise linear. A generalization of the optimal M-term approximation result (6) for piecewise Hölder smooth images with Hölder exponent $\alpha > 1$ has been developed in [18]. In this case, one needs to generalize the idea of a piecewise linear path vector curve to a smooth path function that satisfies, besides the region condition and the diameter condition, a third condition called *path smoothness condition*, see [18]. More precisely, let us consider a domain $\Omega \subset [0,1]^2$ with a sufficiently smooth boundary and a disjoint partition Ω_i of Ω with smooth boundaries of finite length. Further, instead of (5), we assume that $F \in L^2(\Omega)$ is a piecewise smooth bivariate function being Hölder smooth of order $\alpha > 1$ in each region Ω_i, $i = 1,\ldots,K$. In order to show the optimal error estimate (6) also for $\alpha > 1$, we need to employ a path function that approximates the values $f^\ell_{p^\ell(j)}$ at the points $x^\ell_{p^\ell(j)}$ being a planar curve that is not only piecewise smooth but smooth of order α inside a region Ω_i with suitably bounded derivatives, see [18], Section 3.2. Particularly, this condition suggests that one should avoid "small angles" in the path curve.

2.5 Numerical Results

We shortly illustrate the performance of the proposed EPWT algorithm for sparse date representattion and data denoising. In Figure 1, we illustrate the application of the EPWT for sparse image representation, see also [17, 25]. The three considered images are of size 256×256. In Algorithm 1, we have used the 7-9 biorthogonal filter bank for the function values, and the lazy filter bank for the grid points, i.e., at each level of the EPWT, we have kept only every other grid point. The path construction from Subsection 2.2.1 is taken, where in (2) the parameters $\varepsilon = \sqrt{2}$ and $\mu = 5$ are employed. The threshold parameter θ in Algorithm 1 is chosen, such that 1000 most significant EPWT wavelet coefficients are kept for the clock image, 700 coefficients are kept for the Lenna image and 200 coefficients are kept for the sail image. Figure 1 shows the reconstructed images, where we compare the results of a tensor-product wavelet compression with the 7-9 biorthogonal filter bank with the results of the EPWT reconstruction, using the same number of wavelet coefficients for the reconstruction in both cases.

Fig. 1: **Top row:** Reconstruction by tensor-product wavelet compression using the 7-9 biorthogonal filter bank with 1000 wavelet coefficients for test image `clock` (PSNR 29.93), 700 coeffs for `Lenna` (PSNR 24.28), and 200 coeffs for `sail` (PSNR 19.58). **Bottom row**: Reconstruction by EPWT wavelet transform using the 7-9 biorthogonal filter bank with 1000 wavelet coefficients for `clock` (PSNR 33.55), 700 coeffs for `Lenna` (PSNR 30.46), 200 coeffs for `sail` (PSNR 27.19).

In a second example we study the denoising behavior of the EPWT approach as described in Subsection 2.3. In Figure 2, we present the noisy pepper image with a PSNR of 19.97 and compare the denoising results of different methods. In particular, we have used the four-pixel denoising scheme based on anisotropic diffusion by Welk et al. [29] with 76 iterations and step size 0.001 providing a PSNR of 28.26. Further, we apply the 7-9 wavelet shrinkage with a PSNR of 24.91 and the 7-9 wavelet shrinkage with cycle spinning using 64 shifts of the image and yielding the PSNR 28.11. Our EPWT denoising approach employing a relaxed path construction as described in Subsection 2.2.1 achieves a PSNR of 29.01 while a random path construction based on the ideas in Subsection 2.2.2 yields the PSNR 27.96. Note that the repeated application of the EPWT shrinkage method can be done in a parallel process. While our proposed EPWT denoising is (due to the path constructions) more expensive than the tensor-product wavelet shrinkage its application is not restricted to rectangular regular grids.

The third example shows the EPWT denoising to a triangular domain taking the approach in Section 2.3, see Figure 3. We use the 7-9 biorthogonal filter bank for the function values, the lazy filter bank for the grid points, and the path construction from Subsection 2.2.1 with $\varepsilon = 1.3$, $\mu = 89$ and threshold $\theta = 89$.

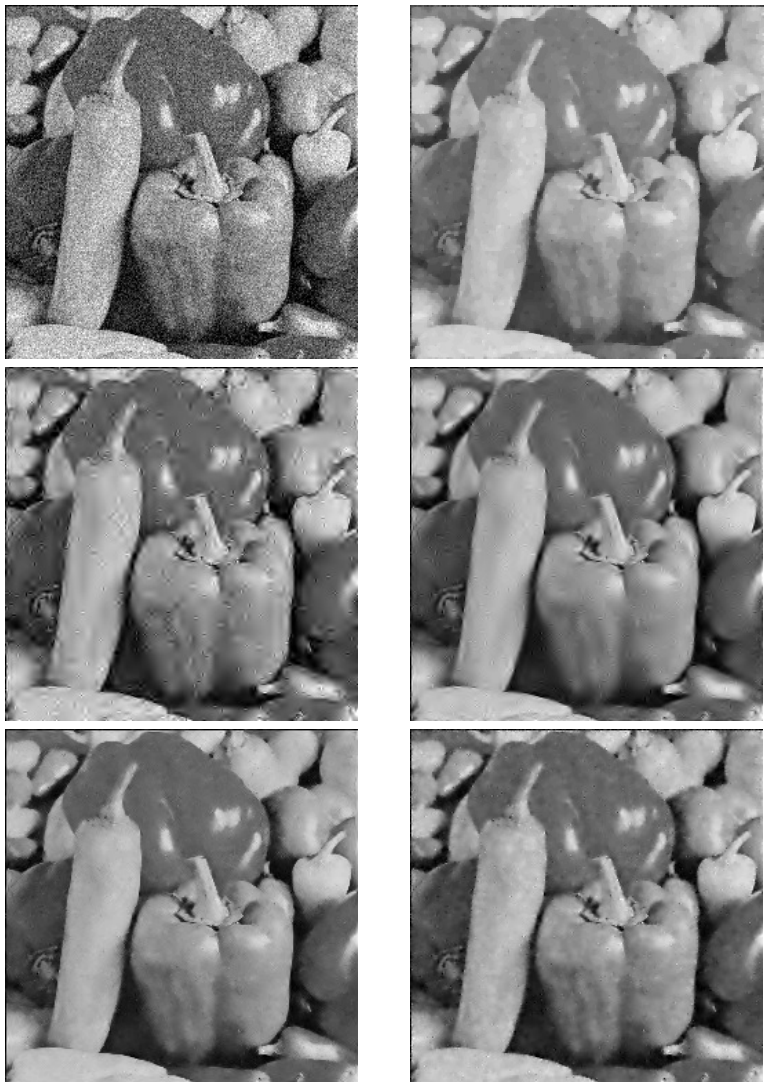


Fig. 2: **Top row:** `Peppers` with additive white Gaussian noise with $\sigma = 25$ (PSNR 19.97) and reconstruction by the Four-Pixel Scheme [29] (PSNR 28.26), **Mid row**: Reconstruction by 2d tensor product wavelet transform using the 7-9 biorthogonal filter bank without (PSNR 24.91) and with cycle spinning (PSNR 28.11) **Bottom row**: Reconstruction by our approach described in Subsection 2.3 using a relaxed path construction with fixed local distances in (2), (PSNR 29.01) and a random path construction based on (4) (PSNR 27.96).

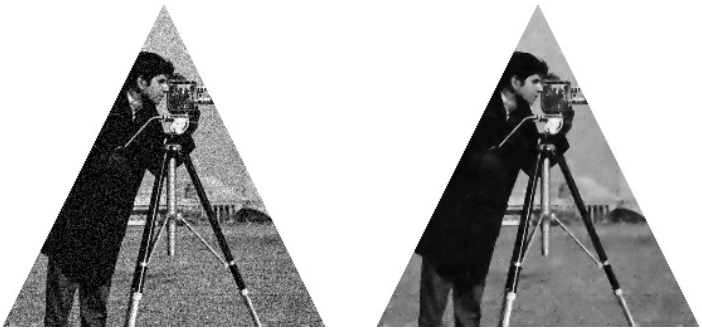


Fig. 3: `Cameraman`. Data with additive white Gaussian noise with $\sigma = 25$ (PSNR 19.98), and EPWT reconstruction using the approach in Section 2.3 (PSNR 26.31).

3 Dimensionality Reduction on High-Dimensional Signal Data

To explain basic concepts on dimensionality reduction, we regard *point cloud data* as a finite family of vectors

$$X = \{x_i\}_{i=1}^m \subset \mathbb{R}^n$$

contained in an n-dimensional Euclidean space. The fundamental assumption is that X lies in $\mathscr{M}$, a low dimensional (topological) *space* embedded in $\mathbb{R}^n$. Therefore, $X \subset \mathscr{M} \subset \mathbb{R}^n$ with $p := \dim(\mathscr{M}) \ll n$. Another ingredient is a parameter domain Ω for $\mathscr{M}$, where Ω is assumed to be embedded in a low dimensional space $\mathbb{R}^d$ with $p \leq d < n$. Moreover, we assume the existence of a homeomorphism (diffeomorphism)

$$\mathscr{A} : \Omega \to \mathscr{M},$$

so that Ω is a homeomorphic (diffeomorphic) copy of $\mathscr{M}$. This concept can then be used for signal analysis in a low dimensional environment. In practice, we can only approximate Ω by a projection

$$P : \mathscr{M} \to \Omega',$$

where Ω' is a homeomorphic copy of Ω. The low dimensional structure representing X is the reduced data $Y = \{y_i\}_{i=1}^m \subset \Omega' \subset \mathbb{R}^d$, according to the following diagram.

$$\begin{array}{ccccc} X & \hookrightarrow & \mathscr{M} & \hookrightarrow & \mathbb{R}^n \\ \downarrow{\scriptstyle P|_X} & & \downarrow{\scriptstyle P} & & \\ Y & \hookrightarrow & \Omega' & \hookrightarrow & \mathbb{R}^d \end{array}$$

Principal component analysis (PCA) is a classical linear projection method. Dimensionality reduction by PCA can be described as an eigenvalue problem, so that PCA can be applied by using the *singular value decomposition* (SVD). More pre-

cisely, in PCA we consider centered data X (i.e., X has zero mean) in matrix form $X \in \mathbb{R}^{n\times m}$. Now the concept of PCA is to construct a linear projection $P : \mathbb{R}^n \to \mathbb{R}^n$, for $\operatorname{rank}(P) = p < n$, with minimal error $\operatorname{err}(P,X) = \sum_{k=1}^m \|x_k - P(x_k)\|$, or, equivalently, with maximal variance $\operatorname{var}(P,X) = \sum_{k=1}^m \|P(x_k)\|^2$. These conditions can in turn be reformulated as an eigenvalue problem, where the p largest eigenvalues of the covariance matrix $XX^T \in \mathbb{R}^{n\times n}$ are sought, cf. [15].

Another classical linear dimensionaly reduction method is *multidimensional scaling* (MDS), which is also relying on an eigendecomposition of data $X \in \mathbb{R}^{n\times m}$. In contrast to PCA, the MDS method constructs a low dimensional configuration of X without using an explicit projection map. More precisely, on input matrix $X \in \mathbb{R}^{n\times m}$, MDS works with the distance matrix $D = (d_{ij})_{i,j=1,\cdots,m}$, of the points in X to compute an optimal configuration of points $Y = (y_1, \cdots, y_m) \in \mathbb{R}^{p\times m}$, with $p \le n$, minimizing the error $\operatorname{err}(Y,D) = \sum_{i,j=1}^m (d_{ij} - \|y_i - y_j\|)^2$. In other words, the low dimensional configuration of points Y preserves the distances of the higher dimensional dataset X approximately.

In the construction of *nonlinear* dimensionality reduction (NDR) methods, we are especially interested in their interaction with signal processing tools, e.g., convolution transforms. When applying signal transforms to the dataset X, one important task is the analysis of the incurred geometrical deformation. To this end, we propose the concept of *modulation maps* and *modulation manifolds* for the construction of particular datasets which are relevant in signal processing and NDR, especially since we are interested in numerical methods for analyzing geometrical properties of the modulation manifolds, with a particular focus on their scalar and mean curvature.

We define a modulation manifold by employing a homeomorphism (or diffeomorphism) $\mathscr{A} : \Omega \to \mathscr{M}$, for a specific manifold Ω, as used in signal processing. The basic objective is to understand how the geometry of Ω is distorted when we transform Ω using a modulation map $\mathscr{A}$. More explicitly, let $\{\phi_k\}_{k=1}^d \subset \mathscr{H}$ be a set of vectors in an Euclidean space $\mathscr{H}$, and $\{s_k : \Omega \to \mathscr{C}_{\mathscr{H}}(\mathscr{H})\}_{k=1}^d$ a family of smooth maps from a manifold Ω to $\mathscr{C}_{\mathscr{H}}(\mathscr{H})$ (the continuous functions from $\mathscr{H}$ into $\mathscr{H}$). We say that a manifold $\mathscr{M} \subset \mathscr{H}$ is a $\{\phi_k\}_{k=1}^d$-*modulated manifold* if

$$\mathscr{M} = \left\{ \sum_{k=1}^d s_k(\alpha)\phi_k,\ \alpha \in \Omega \right\}.$$

In this case, the map $\mathscr{A} : \Omega \to \mathscr{M}$, $\alpha \mapsto \sum_{k=1}^d s_k(\alpha)\phi_k$, is called *modulation map*.

To make one prototypical example (cf. [9]), we regard a map of the form

$$\mathscr{A}(\alpha)(t_i) = \sum_{k=1}^d \phi_k(\alpha_k t_i), \quad \alpha = (\alpha_1, \ldots, \alpha_d) \in \Omega, \quad \{t_i\}_{i=1}^n \subset [0,1],$$

for a set of band-limited functions $\{\phi_k\}_{k=1}^d$ in combination with a finite set of uniform samples $\{t_i\}_{i=1}^n \subset [0,1]$.

Now we use the same notation for the band-limited functions ϕ_k and the above mentioned vector of sampling values $\{\phi_k(t_i)\}_{i=1}^n$, as this is justified by the *Whittaker-Shannon interpolation formula* as follows.

As the support of the band-limited functions ϕ_k is located in $[0,1]$, the Whittaker-Shannon interpolation formula allows us to reconstruct each ϕ_k exactly from the finite samples $(\phi_k(t_i))_{i=1}^n \in \mathbb{R}^n$. This in turn gives a one-to-one relation between the band-limited functions $\phi_k : [0,1] \to \mathbb{R}$ and the vectors $(\phi_k(t_i))_{i=1}^n \in \mathbb{R}^n$. Note that the maps $s_k(\alpha)$ are in our example given by $s_k(\alpha)\phi_k(t_i) = \phi_k(\alpha_k t_i)$. In other words, we use the (continuous) map $s_k(\alpha)$, $f(t) \mapsto f(\alpha_k t)$, as the scaling by factor α_k, being the k-th coordinate of vector $\alpha \in \Omega \subset \mathbb{R}^d$.

To explain our analysis of the geometric distortions incurred by $\mathscr{A}$, we restrict ourselves to the case $d = 3$ and $\Omega \subset \mathbb{R}^3$ with $\dim(\Omega) = 2$. We compute the scalar curvature of $\mathscr{M}$ from the parametrization of Ω and the modulation map $\mathscr{A}$ by the following algorithm [9].

Algorithm 3 *On input parametrization $\alpha = (\alpha_j(\theta_1,\theta_2))_{j=1}^d$ of Ω and band-limited functions $\{\phi_j\}_{j=1}^d$ that are generating the map $\mathscr{A}$, perform the following steps.*

(1) *Compute the Jacobian matrices J_α;*

(2) *Compute the metric tensor* $g_{ij} = \sum_{\ell=1}^{n} t_\ell^2 \sum_{r,q=1}^{d} \left(\frac{d\phi_r}{dt}(\alpha_r t_\ell) \frac{d\phi_q}{dt}(\alpha_q t_\ell) \frac{\partial \alpha_r}{\partial \theta_i} \frac{\partial \alpha_q}{\partial \theta_j} \right)$;

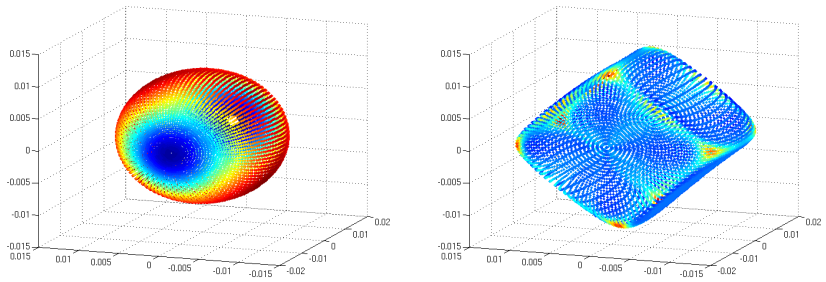
(3) *Compute the Christoffel symbols* $\Gamma_{ij}^k = \frac{1}{2} \sum_{\ell=1}^{p} \left(\frac{\partial g_{j\ell}}{\partial x_i} + \frac{\partial g_{i\ell}}{\partial x_j} - \frac{\partial g_{ij}}{\partial x_\ell} \right) g^{\ell k}$;

(4) *Compute the tensors* ${R^\ell}_{ijk} = \sum_{h=1}^{p} (\Gamma_{jk}^h \Gamma_{ih}^\ell - \Gamma_{ik}^h \Gamma_{jh}^\ell) + \frac{\partial \Gamma_{jk}^\ell}{\partial x_i} - \frac{\partial \Gamma_{ik}^\ell}{\partial x_j}$;

(5) *Compute the scalar curvature* $S = \sum_{i,j=1}^{p} g^{ij} R_{ij}$, *where* $R_{ij} = \sum_{k,\ell=1}^{p} g^{k\ell} R_{kij}$.

Output: *The scalar curvature S of $\mathscr{M} = \mathscr{A}(\Omega)$.*

For further details concerning the construction of Algorithm 3, we refer to [9].



(a) **(b)**

Fig. 4: (a) A sphere Ω whose colors represent the scalar curvature of $\mathscr{M} = \mathscr{A}(\Omega)$, (b) PCA projection of $\mathscr{M} = \mathscr{A}(\Omega)$ with Gaussian curvature represented by colors.

4 Audio Signal Separation and Signal Detection

In many relevant applications of signal processing there is an increasing demand for effective methods to estimate the components from a mixture of acoustic signals. In recent years, different decomposition techniques were developed to do so, including *independent subspace analysis* (ISA), based on *independent component analysis* (ICA), see [2, 6, 27], and *non-negative matrix factorization* (NNMF), see [7, 24, 28]. The computational complexity of these methods, however, may be very large, in particular for real-time computations on audio signals. In signal separation, dimensionality reduction methods are used to first reduce the dimension of the data obtained from a time-frequency transform, e.g., *short time Fourier transform* (STFT), before the reduced data is decomposed into different components, each assigned to one of the source signals. For the application of dimensionality reduction in combination with NNMF, however, *non-negative* dimensionality reduction methods are essentially required to guarantee non-negative output data from non-negative input data (e.g., a non-negative spectrogram from the STFT). For the special case of PCA, a suitable rotation map is constructed in [13] for the purpose of back-projecting the reduced data to the positive orthant of the Cartesian coordinate system, where the sought rotation is given by the solution of a constraint optimization problem in a linear subspace of orthogonal matrices.

In this section, we evaluate different decomposition methods for signal separation in combination with the non-negative PCA projection from [13]. The basic steps of our method are illustrated in Figure 5.

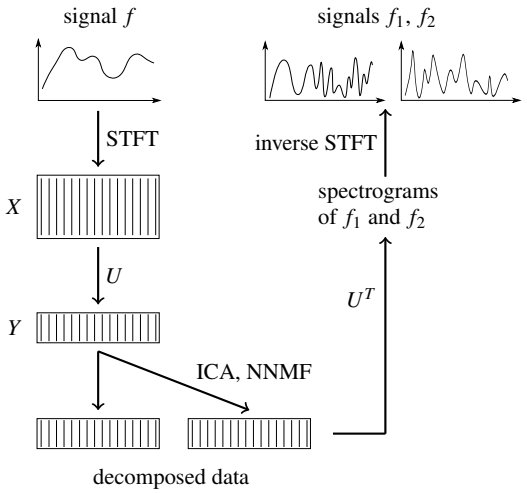


Fig. 5: Signal separation with dimensionality reduction.

To explain how we use PCA, let $U \in \mathbb{R}^{D\times d}$ be an orthogonal projection, satisfying $Y = U^T X$, being obtained by the solution of the minimization problem

$$\min_{\tilde{U}^T\tilde{U}=I} \sum_{k=1}^{n} \left\| x_k - \tilde{U}\tilde{U}^T x_k \right\|_2 . \tag{7}$$

The solution of (7) is given by the maximizer of the variance $\mathrm{var}(Y)$ of Y, as given by the trace of YY^T. This observation allows us to reformulate the minimization problem in (7) as an equivalent maximization problem,

$$\max_{\tilde{U}^T\tilde{U}=I} \mathrm{tr}(\tilde{U}^T X X^T \tilde{U}), \tag{8}$$

where the maximizer U of $\mathrm{var}(Y)$ is given by a matrix U whose d columns contain the eigenvectors of the d largest eigenvalues of the covariance matrix XX^T.

For further processing the data in a subsequent decomposition by NNMF, the data matrix Y is essentially required to be *non-negative*. Note, however, that even if the data matrix X (obtained e.g., by STFT) may be non-negative, this is not necessarily the case for the components of the reduced data matrix Y. Therefore, we reformulate the maximization problem in (8) by adding a non-negativity constraint:

$$\max_{\substack{\tilde{U}^T\tilde{U}=I \\ \tilde{U}^T X \geq 0}} \mathrm{tr}(\tilde{U}^T X X^T \tilde{U}). \tag{9}$$

Note that this additional restriction transforms the simple PCA problem (8) into a much more difficult non-convex optimization problem (9) with many local solutions, for which (in general) none of the solutions is known analytically.

We tackle this fundamental problem as follows. We make use of the fact that the input data set X is *non-negative*, before it is projected onto a *linear* subspace, with the perception that there exists a rotation of the low-dimensional data set Y into the non-negative orthant. Indeed, as proven in [13], such a rotation map exists, which motivates us to split the *non-negative PCA* (NNPCA) problem (9) into a PCA part and a rotation part. This, in turn, gives rise to seek for a general construction of a rotation matrix W satisfying $WU^T X \geq 0$.

To further explain our splitting approach, recall that we already know the solution U of the PCA part. Since the rotation matrix W is orthogonal, it does not affect the value of the NNPCA cost functional. Now, in order to determine the rotation matrix W, we consider solving an auxiliary optimization problem on the set of orthogonal matrices $SO(d)$, i.e., we minimize the cost functional

$$J(\tilde{W}) = \frac{1}{2}\sum_{i,j} \left[\left(\tilde{W}U^T X\right)_- \right]_{ij}^2 \qquad \text{where } [Z_-]_{ij} = \begin{cases} z_{ij} & \text{if } z_{ij} < 0, \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

as this was proposed in [22] in the context of ICA. However, we cannot solve this optimization problem directly by an additive update algorithm, since the set of rotation matrices $SO(d)$ is not invariant under additions. But an elegant way to minimize the

cost functional J in (10) uses the Lie-group structure of $SO(d)$ to transfer the problem into an optimization problem on the Lie-algebra of skew-symmetric matrices $\mathfrak{so}(d)$. Due to the vector space property of $\mathfrak{so}(d)$, standard methods can be applied to find the minimum (see [10, 12, 22] for details).

4.1 Decomposition Techniques

There are different methods for the decomposition of the (reduced) spectrogram Y. Among them, independent component analysis (ICA) and non-negative matrix factorization (NNMF) are commonly used. In either case, for the application of ICA or NNMF, we assume the input data Y to be a linear mixture of source terms s_i, i.e.,

$$Y = AS, \tag{11}$$

where $A \in \mathbb{R}^{d\times r}$ and $S \in \mathbb{R}^{r\times n}$ are unknown. For the estimation of A and S we need specific additional assumptions to balance the disproportion of equations and unknowns in the factorization problem (11).

4.1.1 Independent Component Analysis (ICA)

The basic assumption of ICA is that the source signals are statistically independent. Furthermore, the data matrix Y is assumed to result from n realizations of a d-dimensional random vector. In order to estimate S, a random variable $\mathscr{S}$ is constructed, whose n realizations yield the columns of the source matrix S. The components of $\mathscr{S}$ are chosen to be as stochastically independent as possible, where the stochastical independence can be measured by the *Kullback-Leibler distance* [5].

In practice, the number of sources is usually unknown. Therefore, we may detect more independent components than the true number of sources. In this case, two or more of the separated components belong to the same source. Thus, the sources are combinations of the independent components. In a subsequent step, the sources are grouped into independent subspaces, each corresponding to one source. Finally, the sources are reconstructed from these multi-component subspaces [2]. This procedure is called *independent subspace analysis* (ISA). The main difficulty of ISA is to identify components belonging to the same multi-component subspace.

4.1.2 Non-Negative Matrix Factorization (NNMF)

The factorization of the given data Y into a mixing matrix A and the source signals (source components) S, i.e., $Y = AS$, could be done by matrix factorization. The data we use for signal separation are obtained by taking the modulus of the signal's STFT, and so the input data is non-negative. Since the source components are assumed to

be spectrograms, too, we assume them to be non-negative as well. Therefore, non-negative matrix factorizations (NNMF) are suitable tools for decomposition.

There are different NNMF algorithms available, all of which are relying on the non-negativity $Y,A,S \geq 0$, where different measures $d(Y,AS)$ for the reconstruction error were proposed [7, 24, 28]. We consider using the generalized *Kullback-Leibler distance* (proposed in [14] and used for decomposing signal data in [28]):

$$d(Y,AS) = \sum_{i,j} Y_{ij} \log \frac{Y_{ij}}{(AS)_{ij}} - Y_{ij} + (AS)_{ij}.$$

4.2 Numerical Results

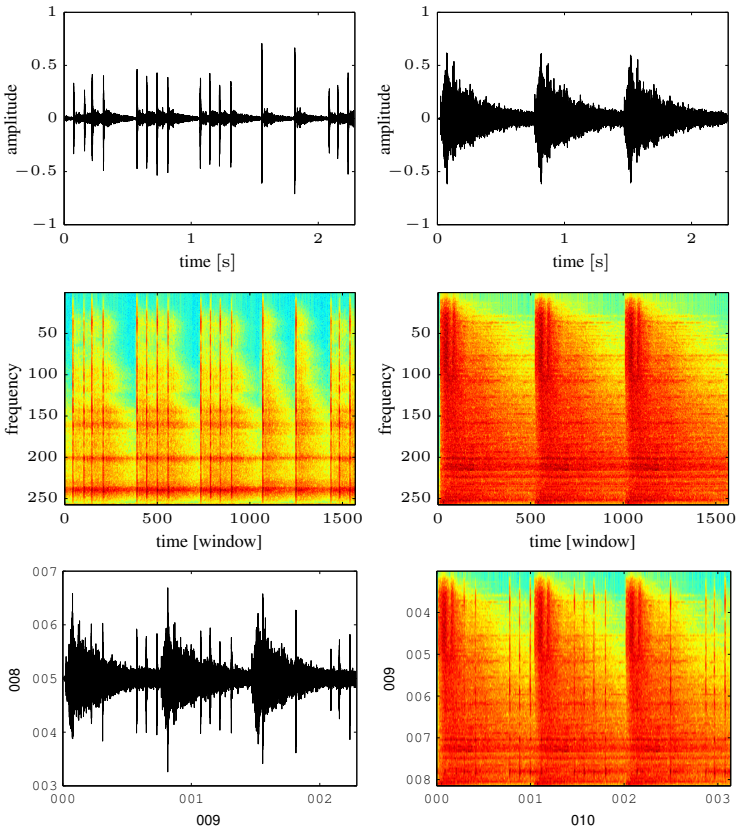


Fig. 6: Two acoustic signals: castanets f_1 (top left), cymbal f_2 (top right), and corresponding spectrograms (2nd row). Signal $f = f_1 + f_2$ and spectrogram (3rd row).

We present one numerical example comparing the decomposition strategies ICA and NNMF. We consider a mixture $f = f_1 + f_2$ of acoustic transient signals, where f_1 is a sequence of castanets and f_2 a cymbal signal, shown in Figure 6, where also the combination $f = f_1 + f_2$ of the two signals is displayed. The spectrograms in these figures are generated with an STFT using a Hamm-window. Since f_2 is a high-energy signal, f has a complex frequency characteristic. Therefore, the extraction of the castanets signal f_1, being active only at a few time steps, is a challenging task.

The obtained separations, resulting from the two different decomposition methods using NNPCA and PCA, respectively, are displayed in Figure 7. Note that both methods, NNMF and ICA, achieve to reproduce the characteristic peaks of the castanets quite well. However, in the case of NNMF strong artifacts of the castanets are visible in the cymbal signal, whereas the separation by ICA is almost perfect.

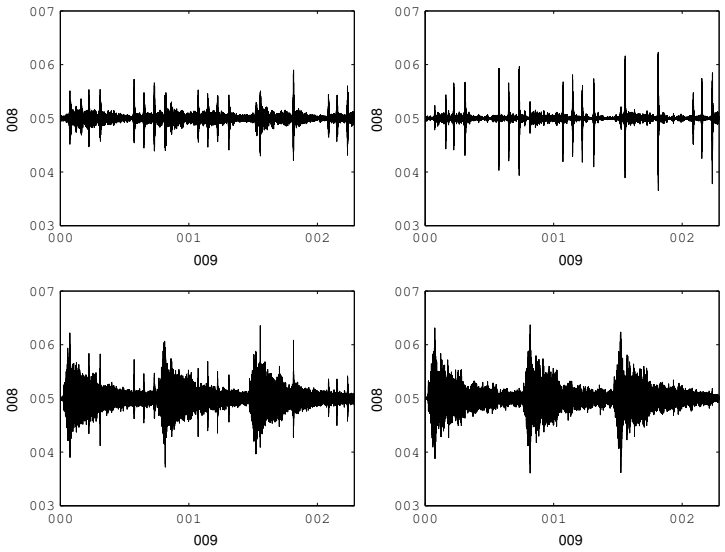


Fig. 7: Signal separation by NNPCA & NNMF (left col.); PCA & ICA (right col.).

Likewise, for the reconstruction of the reduced signal, the combination of PCA and ICA provides an almost complete reproduction of the original signal f (see Figure 8). Merely at time steps where a high amplitude of the cymbal exactly matches the peaks of the castanets, a correct separation is not quite achieved. As for the NNMF, the spectrogram in Figure 8 shows that information is being lost.

We finally remark that for signal separation *without* dimensionality reduction, NNMF is competitive to ICA (see e.g. [28]). This indicates that our use of NNPCA in combination with NNMF could be improved. Further improvements could be achieved by the use of more sophisticated (nonlinear) dimensionality reduction methods. On the other hand, this would lead to a much more complicated construc-

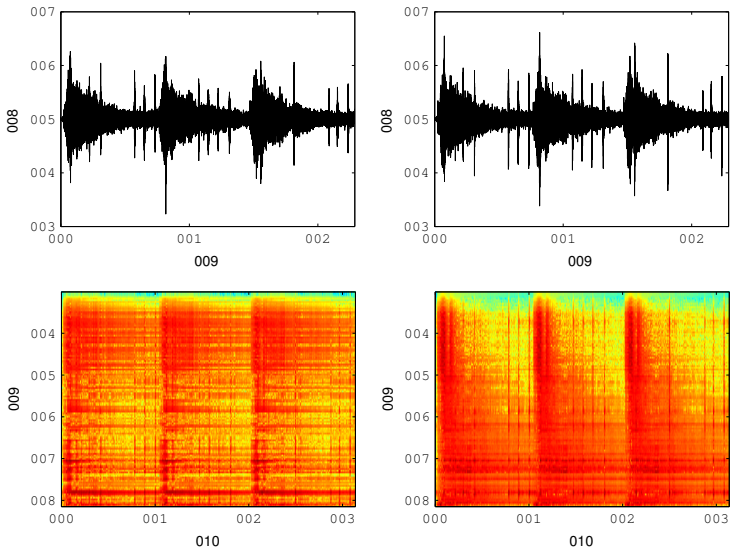


Fig. 8: Reconstruction of f as sum of the decomposed f_i by using NNPCA & NNMF (left column) and by using PCA & ICA (right column).

tion of the inverse transform, as required for the back-projection of the data. We defer these points to future research. Nevertheless, although PCA is only a *linear* projection method, our numerical results of this section, especially those obtained by the combination of PCA and ICA, are already quite promising.

Acknowledgements This work is supported by the priority program DFG-SPP 1324 of the Deutsche Forschungsgemeinschaft (DFG), through projects PL 170/13-2 and IS 58/1-2.

References

1. G. Carlsson: Topology and data. *Bull. Amer. Math. Soc.* **46**(2), 2009, 255–308.
2. M. A. Casey and A. Westner: Separation of mixed audio sources by independent subspace analysis. In: *Proceedings of the International Computer Music Conference*, Berlin, 2000.
3. A. Cohen, I. Daubechies, and J. Feauveau, Biorthogonal bases of compactly supported wavelets, *Comm. Pure Appl. Math.* **45**, 1992, 485–560.
4. R.R. Coifman and D.L. Donoho: Translation-invariant de-noising. In: *Wavelets and Statistics*, A. Antoniadis and G. Oppenheim (eds.), Springer, New York, 1995, 125–150.
5. P. Comon: Independent component analysis, a new concept? *Signal Processing* **36**(3), 1994, 287–314.
6. D. Fitzgerald, E. Coyle, and B. Lawlor: Sub-band independent subspace analysis for drum transcription. In: *Proceedings of the 5th International Conference on Digital Audio Effects (DAFX-02)*, Hamburg, Germany, 2002.

7. D. FitzGerald, M. Cranitch, and E. Coyle: Non-negative tensor factorisation for sound source separation. In: *Proceedings of Irish Signals and Systems Conference*, Dublin, Ireland, 2005, 8–12.
8. M. Guillemard: *Some Geometrical and Topological Aspects of Dimensionality Reduction in Signal Analysis*. PhD thesis, University of Hamburg, 2011, `ftp://ftp.math.tu-berlin.de/pub/numerik/guillem/prj2/mgyDiss.pdf`.
9. M. Guillemard and A. Iske: Curvature analysis of frequency modulated manifolds in dimensionality reduction. Calcolo **48**(1), 2011, 107–125.
10. B.C. Hall: *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Graduate Texts in Mathematics, vol. 222, Springer, New York, 2004.
11. D. Heinen and G. Plonka: Wavelet shrinkage on paths for denoising of scattered data. *Result. Math.* **62**(3), 2012, 337–354.
12. A. Iserles, H.Z. Munthe-Kaas, S. Nørsett, and A. Zanna: Lie-group methods. *Acta Numerica*, 2000, 215–365.
13. S. Krause-Solberg and A. Iske: Non-negative dimensionality reduction in signal separation. Preprint, University of Hamburg, 2013.
14. D.D. Lee and H.S. Seung: Algorithms for non-negative matrix factorization. In: *Advances in Neural Information Processing Systems*, vol. 13, MIT Press, 2000, 556–562.
15. J.A. Lee and M. Verleysen: *Nonlinear Dimensionality Reduction*. Springer, 2010.
16. S. Mallat: *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, 1999.
17. G. Plonka: The easy path wavelet transform: a new adaptive wavelet transform for sparse representation of two-dimensional data. *Multiscale Modelling Simul.* **7**, 2009, 1474–1496.
18. G. Plonka, A. Iske, and S. Tenorth: Optimal representation of piecewise Hölder smooth bivariate functions by the easy path wavelet transform. *J. Approx. Theory* **176**, 2013, 42–67.
19. G. Plonka and D. Roşca: Easy path wavelet transform on triangulations of the sphere. *Mathematical Geosciences* **42**(7), 2010, 839–855.
20. G. Plonka, S. Tenorth, and A. Iske: Optimally sparse image representation by the easy path wavelet transform. *Int. J. Wavelets Multiresolut. Inf. Process.* **10**(1), 2012, 1250007 (20 pages).
21. G. Plonka, S. Tenorth, and D. Roşca: A hybrid method for image approximation using the easy path wavelet transform. *IEEE Trans. Image Process.* **20**(2), 2011, 372–381.
22. M.D. Plumbley: Geometrical methods for non-negative ICA: manifolds, Lie groups and toral subalgebras. *Neurocomputing* **67**, 2005, 161–197.
23. I. Ram, M. Elad, and I. Cohen: Generalized tree-based wavelet transform. *IEEE Trans. Signal Process.* **59**(9), 2011, 4199–4209.
24. P. Smaragdis and J.C. Brown: Non-negative matrix factorization for polyphonic music transcription. In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003, 177–180.
25. S. Tenorth: *Adaptive Waveletmethoden zur Approximation von Bildern*. PhD thesis, University of Göttingen, 2011, `http://hdl.handle.net/11858/00-1735-0000-0006-B3E7-A`.
26. 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.
27. C. Uhle, C. Dittmar, and T. Sporer: Extraction of drum tracks from polyphonic music using independent subspace analysis. In: *Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)*, Nara, 2003, 843–848.
28. T. Virtanen: Monaural sound source separation by non-negative matrix factorization with temporal continuity and sparseness criteria. *IEEE Trans. on Audio, Speech, and Language Process.* **15**(3), 2007, 1066–1074.
29. M. Welk, J. Weickert, and G. Steidl: A four-pixel scheme for singular differential equations. In: *Scale-Space and PDE Methods in Computer Vision*, Berlin, 2005, 610–621.