Multiscale local polynomial smoothing in a lifted pyramid for non-equispaced data

Maarten Jansen
Université Libre de Bruxelles, departments of Mathematics and Computer Science

June 2012

Abstract

This paper integrates Burt-Adelson's Laplacian pyramids with lifting schemes for the construction of slightly redundant decompositions. These decompositions implement multiscale smoothing on possibly non-equidistant point sets. Thanks to the slight redundancy and to the smoothing operations in the lifting scheme, the proposed construction unifies sparsity of the analysis, smoothness of the reconstruction and stability of the transforms. The design of these decomposition requires pairs of non-stationary filters, adapted to each other and to the locations of the observations at hand. This paper provides a general algorithm for the construction of such band-limited pairs of filter matrices.

1 Introduction

Multiscale decompositions for non-equispaced data have always been a non-trivial extension of the classical, equispaced, dyadic wavelet transforms. In the non-equidistant case, basis functions cannot be dilations and translations of a single mother or father function. Indeed, the construction of these basis functions, such as the one used in the lifting scheme [17, 18], starts off from the locations of the observations. When these observations are not equidistant, it is impossible to define functions that are shifts of each other, neither is there a precise notion of scale, because the distance between adjacent observations is not fixed. As a consequence, important parts of the theory supporting wavelet analyses need to be reworked. Other constructions, including [3, 13, 1], stay within the classical wavelet decomposition, but processing irregular samples may require preprocessing or may be suboptimal from the computational point of view.

The missing parts in the theory behind non-equispaced wavelet decompositions are related to problematic behavior of some of these decompositions in practical situations. Four issues in the design of a decomposition can be identified. The first issue is perfect reconstruction (PR). The transform should be invertible. In general, this condition is fairly easy to satisfy, especially in a classical lifting scheme, where each operation is readily invertible. In other cases, such as the variant on lifting adopted in this

paper, it is an algebraic matter that can be solved scale by scale. The second issue is sparsity of the analysis. In a lifting scheme construction, this objective is fairly easy to control. The third issue is the numerical condition of the decomposition and reconstruction. The condition can be controlled at two levels. The first is scale-to-scale condition, i.e., the error propagation when proceeding from a fine scale representation to one scale coarser with fine scale details. This one-step condition can be managed without too much difficulties. More problematic is the overall numerical condition, that is, the numerical condition when decomposing data at arbitrarily fine scale into details at many scales. The fourth issue is smoothness of the basis functions at the synthesis side. These basis functions can be retrieved using a numerical procedure called subdivision. This subdivision scheme, further explained below, essentially finds the basis function by gradual refinement at successive scales. With observations at regular locations, the refinement takes place at midpoints in each step, and the infinite refinement scheme can be analyzed completely by looking at a single step, resulting in a two-scale equation. When the refinement is irregular, the two-scale equation itself becomes scale-dependent and therefore difficult to analyze.

Throughout this paper, smoothness at reconstruction and numerical condition will turn out to be somehow contradictory objectives. The two objectives can be reconciled by replacing the critically downsampled lifting scheme by a redundant alternative. The result can be seen as a lifted and non-equispaced version of Burt-Adelson's Laplacian pyramid [2]. Lifted pyramids have been proposed in earlier work [9] as well, in a different context, however, and with different design objectives. As the transform is redundant, the inverse transform is not unique. We propose a reconstruction with an additional smoothing, thereby generalizing results for equispaced data [8, 14]. As the pyramid filter operations in this paper are non-stationary, due to the non-equispaced nature of the observations, special attention is paid to the grid-adaptive design of these sparse filter matrices, satisfying perfect reconstruction, and leading to smooth reconstructions.

The paper is organized as follows. Section 2 summarizes the principle ideas of the lifting scheme, whose tools will be adopted in the subsequent sections. Section 3 replaces interpolation by smoothing as one of the basic techniques used in a lifting scheme. Multiscale smoothing should be more robust to numerical errors and it should lead to smoother reconstructions. The section explains that the incorporation of smoothing into lifting requires the transform to become overcomplete, which leads to a nonequispaced version of a Laplacian pyramid. The remainder of the section is devoted to design properties, including perfect reconstruction and vanishing moments, and their translation into design conditions. Section 4 illustrates the new multiscale decompositions with an experiment on denoising, followed by a conclusion in Section 5.

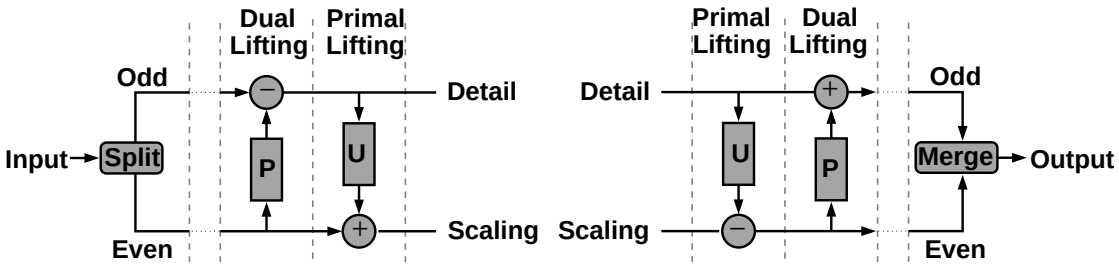

Figure 1: General lifting scheme and its inverse. The lifting scheme depicted here is an implementation of a binary filter bank. There are two sorts of lifting steps, primal and dual, operating on the detail and the approximation branches, respectively. The dots between the split stage and the actual lifting steps represent the possibility of having a longer sequence than just two of those steps. The inverse transform follows immediately by reflection of the scheme and switching plus- and minus-signs.

2 The lifting scheme

2.1 Lifting steps

The lifting scheme is an implementation of a filter bank. All filter banks used in classical wavelet theory can be decomposed into a sequence of lifting steps [6]. Those lifting steps operate between two subsets of the input data, as illustrated in Figure 1. In this section, we focus on a filter bank that operates at scale j. That implies that we suppose that its input comes from the output of a previous filter bank at scale $j+1$, while one of its output branches is further processed by a filter bank at coarser scale $j-1$. In a lifting implementation, the input of the filter bank is partitioned into *even* and *odd* indexed subsets. That is, if $\mathcal{J}_j$ is the set of indices into the data vector $\boldsymbol{s}_j$, then lifting starts by splitting it into $\mathcal{J} = e \cup o$, with $e \cap o = \emptyset$. This expression does not impose a strict even-odd alternated partitioning. Other splits are possible, and sometimes also recommended, for instance if the observations are at irregular time points [19, 20]. The actual lifting operations start after the split and appear in two kinds, depending on the direction they work. Dual lifting operates from the even onto the odd branch, and can be interpreted as a prediction of the odd indexed values, based in the evens. The difference between observed value and neighbor based prediction is typically small, and acts as a wavelet or detail coefficient at the current scale. Primal lifting operates on the odd branch, and can be interpreted as updating the even values before proceeding to the next, coarse scale. The update can be seen as anti-aliasing in signal processing terms, as numerical stabilization, or as variance reduction, smoothing in statistical terms.

A general lifting scheme consists of a sequence of mostly alternating dual and primal lifting steps. As an illustration of the formalism, the remainder of this section elaborates the case of a single dual step followed by a single primal step. Denote by P_j the dual (prediction) lifting operation at scale j. It is a $n_{j,o} \times n_{j,e}$ matrix. Similarly, the $n_{j,e} \times n_{j,o}$ matrix U_j stands for the primal (update) lifting operation.

Then, the forward lifting transform is

$$\begin{aligned} \boldsymbol{w}_j &= \boldsymbol{s}_{j+1,o} - \mathrm{P}_j \cdot \boldsymbol{s}_{j+1,e} && (1) \\ \boldsymbol{s}_j &= \boldsymbol{s}_{j+1,e} + \mathrm{U}_j \cdot \boldsymbol{w}_j && (2) \end{aligned}$$

The inverse of the operation follows immediately by inversion of the expressions above.

$$\begin{aligned} \boldsymbol{s}_{j+1,e} &= \boldsymbol{s}_j - \mathrm{U}_j \cdot \boldsymbol{w}_j && (3) \\ \boldsymbol{s}_{j+1,o} &= \boldsymbol{w}_j + \mathrm{P}_j \cdot \boldsymbol{s}_{j+1,e} && (4) \end{aligned}$$

2.2 Underlying basis transform

In order to associate basis functions with this transform, we define a function

$$f_{j+1}(t) = \Phi_{j+1}(t) \cdot \boldsymbol{s}_{j+1},$$

where $\Phi_{j+1}(t)$ is a vector of *primal* basis functions, i.e., $\Phi_{j+1}(t) = [\dots \varphi_{j+1,k}(t) \dots]$. The coefficients $\boldsymbol{s}_{j+1}$ are thought to be inner products of $f_{j+1}(t)$ with a dual basis $\widetilde{\Phi}_{j+1}$, i.e.,

$$\boldsymbol{s}_{j+1} = \widetilde{\Phi}_{j+1}^T \cdot f_{j+1}(t),$$

this expression being a slight abuse of notation (inspired by its discrete analogue), denoting the vector of inner products $[\dots \langle \tilde{\varphi}_{j+1,k}, f \rangle \dots]^T$.

One step of a (lifted) filter bank transforms this decomposition into another one

$$f_{j+1}(t) = \Phi_j(t) \cdot \boldsymbol{s}_j + \Psi_j(t) \cdot \boldsymbol{w}_j,$$

with

$$\boldsymbol{s}_j = \widetilde{\Phi}_j^T \cdot f_{j+1}(t), \quad \boldsymbol{w}_j = \widetilde{\Psi}_j^T \cdot f_{j+1}(t), \tag{5}$$

The functions in $\Psi_j(t)$ are named primal or synthesis wavelets, as they are used upon reconstruction of a signal. Likewise, the functions in $\widetilde{\Psi}_j$ are dual or analysis wavelets, as they are used to decompose a function into a (primal) lifting basis.

2.3 Dual basis equations

Plugging in (5) into (1) leads to the dual wavelet equations. The function $f_{j+1}(t)$ can be omitted, since the expression would hold for *any* $f_{j+1}(t)$.

$$\widetilde{\Psi}_j^T = \widetilde{\Phi}_{j+1,o}^T - \mathrm{P}_j \cdot \widetilde{\Phi}_{j+1,e}^T$$

Taking the transpose brings us to

$$\widetilde{\Psi}_j = \widetilde{\Phi}_{j+1,o} - \widetilde{\Phi}_{j+1,e} \cdot \mathrm{P}_j^T \tag{6}$$

The two-scale equation follows similarly from (2)

$$\widetilde{\Phi}_j^T = \widetilde{\Phi}_{j+1,e}^T + \mathrm{U}_j \cdot \widetilde{\Psi}_j^T$$

Taking transpose and plugging in (6) leads to

$$\widetilde{\Phi}_j = \widetilde{\Phi}_{j+1,e} \cdot (I - \mathrm{P}_j^T \mathrm{U}_j^T) + \widetilde{\Phi}_{j+1,o} \cdot \mathrm{U}_j^T \tag{7}$$

2.4 Subdivision and primal basis, design of dual lifting steps

The primal basis functions follow from a different analysis, namely subdivision. We start from the observation that $\varphi_{j,k} = \Phi_j \cdot \boldsymbol{\delta}_k + \Psi_j \cdot \mathbf{0}$ with $\boldsymbol{\delta}_k$ the Kronecker-delta at index k, i.e., $\delta_{k,i} = 0$ for $i \neq k$ and $\delta_{k,k} = 1$.

Feeding the inverse step with the inputs $\boldsymbol{s}_j = \boldsymbol{\delta}_k$ and $\boldsymbol{w}_j = \mathbf{0}$ leads to

$$\begin{aligned} \boldsymbol{s}_{j+1,e} &= \boldsymbol{\delta}_k \\ \boldsymbol{s}_{j+1,o} &= \mathrm{P}_j \cdot \boldsymbol{\delta}_k \end{aligned}$$

This allows the conclusion

$$\varphi_{j,k} = \Phi_{j+1,e} \cdot \boldsymbol{\delta}_k + \Phi_{j+1,o} \cdot \mathrm{P}_j \cdot \boldsymbol{\delta}_k.$$

We can repeat the same argument for all k, i.e., we can feed the inverse step with $I = [\ldots \boldsymbol{\delta}_k \ldots]$, to arrive at the two-scale equation (also known as refinement equation)

$$\Phi_j = \Phi_{j+1,e} + \Phi_{j+1,o} \cdot \mathrm{P}_j. \tag{8}$$

If the input is $\boldsymbol{w}_j = \boldsymbol{\delta}_k$ and $\boldsymbol{s}_j = \mathbf{0}$, then we conclude

$$\psi_{j,k} = \Phi_{j+1,e} \cdot (-\mathrm{U}_j \boldsymbol{\delta}_k) + \Phi_{j+1,o} \cdot (\boldsymbol{\delta}_k - \mathrm{P}_j \mathrm{U}_j \boldsymbol{\delta}_k),$$

which, after expansion to $I = [\ldots \boldsymbol{\delta}_k \ldots]$, becomes the wavelet equation

$$\Psi_j = -\Phi_{j+1,e} \mathrm{U}_j + \Phi_{j+1,o} \cdot (I - \mathrm{P}_j \mathrm{U}_j). \tag{9}$$

Solving the two-scale and wavelet equations reveals the underlying synthesis basis, and thus the properties of any reconstruction in such basis. An important numerical solver is the actual subdivision scheme. It consists of iterated application of the two-scale equation in order to further refine the outcome

at scale $j+1$ in terms of basis functions at increasingly finer scales. That is, we write Φ_j or Ψ_j as linear combinations of Φ_{j+1}, then Φ_{j+2} and so on, by repeated application of (8) on its own output, and with incrementing index j. The hope is that the scaling functions Φ_j "converge" to infinitely narrow Dirac impulses, so that the sequence of coefficients in the linear combination reflect the properties of the original basis functions Φ_j or Ψ_j.

It should be noted that the refinement for wavelet functions in Ψ_j involve the wavelet equation only once, followed by an infinite refinement through the two-scale equation. The wavelet equation writes wavelet functions as a simple linear combination of scaling functions, and thus it has no fundamental impact on *individual* limiting properties of the functions, in particular on their smoothness. The whole of wavelet equations at each scale affects the *joint* properties of the basis, such as its numerical stability (see Section **??**).

Besides on the smoothness of the primal basis, the dual lifting step has an immediate impact on the sparsity of the transform. While smoothness of subdivision is hard to analyze, the sparsity is fairly easy to control. The dual lifting is a prediction that can be designed such that certain types of smooth functions are predicted exactly, leading to all zero detail coefficients. These smooth functions are mostly polynomials: the number of dual vanishing moments is the largest integer $\tilde{n}$ for which $\int_{-\infty}^{\infty} \widetilde{\Psi}_j^T(t) t^{\tilde{n}-1} dt = \mathbf{0}$. It corresponds to the degree of polynomials with detail coefficients zero. All functions that can be approximated well by piecewise polynomials then decompose into wavelet coefficients that are close to zero, except at the locations of jumps. The construction of the prediction is often based on interpolation in the evens, for reasons that become clear in Section 3.

Fixing vanishing moments in a prediction operating on even indexed points and evaluated in odd indexed points, naturally involves the location of those points. The irregularity of the observations is taken into account. This has the benefit that the basis functions will be grid-adaptive. On the other hand, the two-scale equation will be grid-adaptive, and so it will be different at each scale. This complicates the convergence analysis of the subdivision scheme. The multitude of possible refinements and arbitrary grids leaves little hope for general results about smoothness of subdivision. Notable exceptions include subdivision by cubic polynomial interpolation [4].

2.5 Design of primal lifting steps, primal moments

Wavelet basis functions are linear combinations of scaling functions, based Define the moments of the primal basis functions as

$$\begin{aligned} M_j^{(p)} &= \int_{-\infty}^{\infty} \Phi_j^T(t) t^p dt \\ O_j^{(p)} &= \int_{-\infty}^{\infty} \Psi_j^T(t) t^p dt, \end{aligned}$$

then we can integrate the two-scale- and wavelet equations

Integrating (8) leads to

$$M_j^{(p)} = M_{j+1,e}^{(p)} + \mathrm{P}_j^T \cdot M_{j+1,o}^{(p)} \tag{10}$$

Integrating (9) leads to

$$O_j^{(p)} = -\mathrm{U}_j^T \cdot M_{j+1,e}^{(p)} + (I - \mathrm{U}_j^T \mathrm{P}_j^T) \cdot M_{j+1,o}^{(p)} \tag{11}$$

Using the former expression to eliminate $M_{j+1,e}^{(p)}$ from the latter, we can write

$$O_j^{(p)} = M_{j+1,o}^{(p)} - \mathrm{U}_j^T \cdot M_j^{(p)}$$

This can be seen as the second step in a lifting scheme whose first step is (10). The first step transforms even fine scaling coefficients into coarse scaling coefficients, so this step is an update. The second step defines the details, so this step is prediction. We thus have an *adjoint* (or conjugate) update-first lifting scheme with prediction operator U_j^T and update operator P_j^T, while the original lifting schema was a prediction-first one.

Design of the update lifting step follows often from imposing vanishing moments in the primal basis, i.e., $O_j^{(p)} = \mathbf{0}$. The resulting expression $M_{j+1,o}^{(p)} = \mathrm{U}_j^T \cdot M_j^{(p)}$ can be seen as a set of equations in unknowns U_j (for *given* P_j, whose design determines $M_j^{(p)}$). Primal vanishing moments contribute to numerical stability. For instance, if the basis functions have no zero integral (i.e., one vanishing moment), then the zero function can be decomposed in a nontrivial way and the decomposition converges in quadratic norm (in L_2) [12]. Without zero integrals, the basis functions cannot be called real wavelets. Such basis is known as a hierarchical basis. Nonlinear processing (such as thresholding) in a hierarchical basis is beneficial, but only in function spaces such as Sobolev spaces that impose more smoothness than L_2 [15]. Higher order primal vanishing moments are less crucial, and may be replaced by other criteria that promote anti-aliasing, numerical stabilization, smoothing and variance reduction.

While the update step is necessary to impose zero integrals, and hence numerical stability in L_2, it may also — inevitably — introduce other forms of numerical instability. In particular, if the grid of data points is very irregular, showing large gaps adjacent to small gaps, then noisy fluctuations on small gaps may be blown up in an interpolating prediction that is evaluated across a large gap. In matrix terms, the prediction operation coefficients are unbounded. The scaling functions that come out of the subdivision have heavy side lobes, which almost overlap with similar lobes in adjacent scaling functions. The two adjacent scaling basis functions are far from orthogonal. An update step that creates zero integral wavelet functions out of these, cannot undo the obliqueness of the basis. Possible remedies consist of avoiding as much as possible strongly inhomogeneous grids for interpolating prediction. This can be done by relaxing the strict even-odd split into a more careful partitioning [19] or by constructing the interpolating prediction on a different subset of the evens if the closest evens to a given odd are too close to each other [20]. Such remedies are limited in the sense that they cannot ideally take care of all possible sample configurations.

3 Smoothing prediction

3.1 Kernel and local polynomial smoothing prediction

In order to bound the prediction coefficients, interpolation can be replaced by smoothing in the prediction. This paper proposes to use kernel smoothing as basic operation in the prediction step. The prediction coefficients $\mathrm{P}_{j,k,\ell}$ at scale j in the expression

$$w_{j,k} = s_{j+1,2k+1} - \sum_{\ell} \mathrm{P}_{j,k,\ell} s_{j+1,2\ell},$$

are filled in by $\mathrm{P}_{j,k,\ell} = P_{j,k}(t_{j+1,2\ell+1}; \boldsymbol{t}_{j+1,e})$, where

$$P_{j,k}(t; \boldsymbol{t}_{j+1,e}) = \frac{K\left(\frac{t-t_{j+1,2k}}{h_{j+1}}\right)}{\sum_{i=1}^{2^j} K\left(\frac{t-t_{j+1,2i}}{h_{j+1}}\right)}, \tag{12}$$

where $K(t)$ is a kernel (a positive function) and h_{j+1} a scale dependent bandwidth. Kernel prediction coefficients constitute a convex combination of the adjacent observations, i.e., the coefficients are positive and sum up to one. Such prediction is numerically stable.

Kernel prediction is a special case of local polynomial prediction, which takes the form $\mathrm{P}_j = [\ldots\ P_j^T(t_{j+1,2k+1}; \boldsymbol{t}_{j+1,e})\ \ldots]^T$ where $P_j(t; \boldsymbol{t}_{j+1,e})$ is a row matrix of length equal to that of $\boldsymbol{t}_{j+1,e}$, depending on variable t and defined by

$$P_j(t; \boldsymbol{t}_{j+1,e}) = \mathrm{T}^{(\tilde{n})}(t) \left({\mathrm{T}^{(\tilde{n})}_{j+1,e}}^T W_{j+1,e}(t) \mathrm{T}^{(\tilde{n})}_{j+1,e} \right)^{-1} \left({\mathrm{T}^{(\tilde{n})}_{j+1,e}}^T W_{j+1,e}(t) \right). \tag{13}$$

In this definition, we use $\mathrm{T}^{(\tilde{n})}(\boldsymbol{t}) = [1\, t\ \ldots\ t^{\tilde{n}-1}]$, with $\tilde{n} \in \{1, 2, \ldots\}$ the order of the prediction, $\tilde{n}-1$ being the polynomial degree. The case $\tilde{n} = 1$ corresponds to kernel prediction. Other notations adopted in (13) include the vector $\boldsymbol{t}^p$, which stands for pointwise exponentiation of a vector $\boldsymbol{t}$. This allows to define the matrix $\mathrm{T}^{(\tilde{n})}_{j+1,e} = [\mathbf{1}\, \boldsymbol{t}_{j+1,e}\ \ldots\ \boldsymbol{t}^{\tilde{n}-1}_{j+1,e}]$. Finally, $W_{j+1,e}(t)$ is a diagonal matrix of weights with elements $(W_{j+1,e})_{kk}(t) = K\left(\frac{t-t_{j+1,2k}}{h_{j+1}}\right)$. Centering and orthogonalization lead to numerically more stable expressions than (13). It should be noted that general local polynomial prediction is not convex. It is, however, possible to replace the weighted least squares formula in (13) by other constructions within the kernel.

Kernel or local polynomial prediction operation has an important drawback. It is not continuous, meaning that

$$\lim_{u \to t_{j,2k}} P_j(u; \boldsymbol{t}_{j+1,e}, \boldsymbol{s}_{j+1,e}) \neq s_{j,2k}.$$

The continuity condition is essential for smooth subdivision, which explains the popularity of interpolation in subdivision schemes. An example of subdivision with smoothing prediction is simulated in Figure 2.

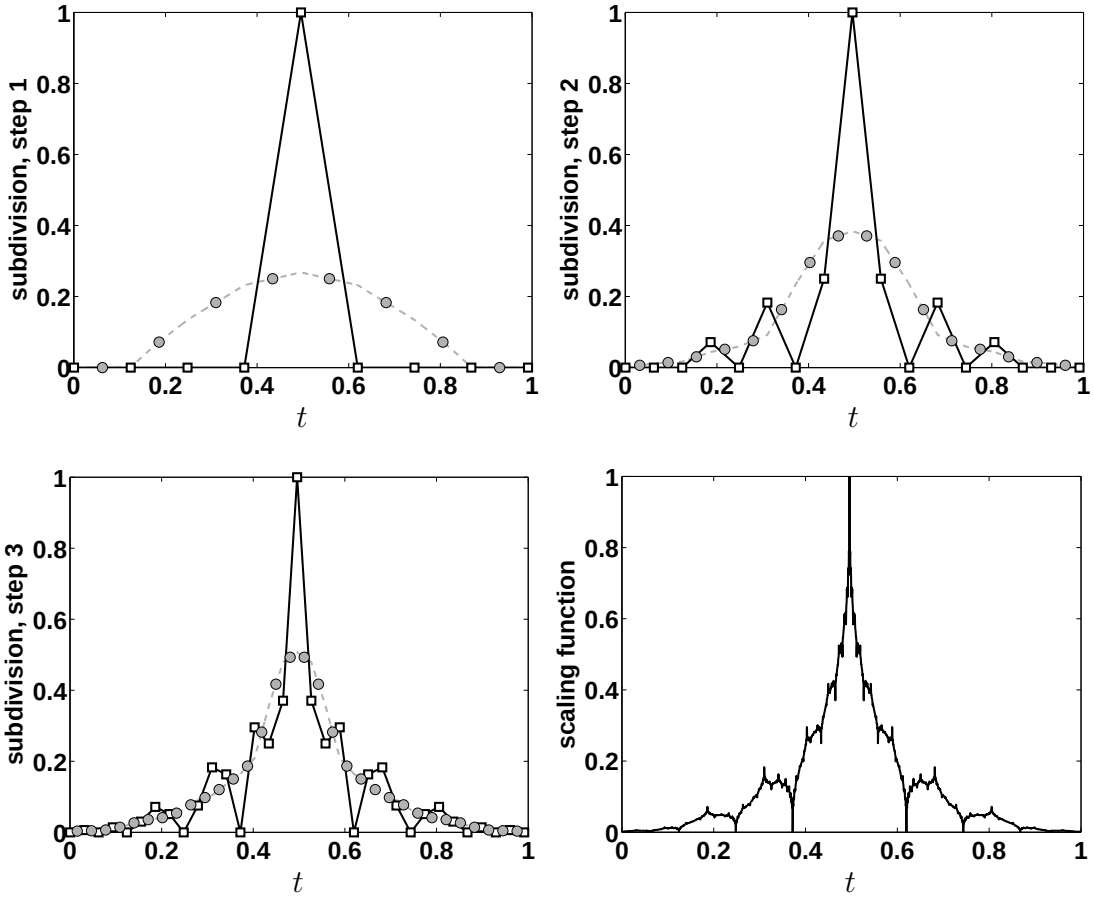

Figure 2: Subdivision steps ($1-2-3-\infty$) with smoothing prediction. We start with a single nonzero scaling coefficient (and all wavelet coefficients are zero). The refinement thus leads to a scaling basis function. But as the predictions (in grey, broken lines) are not interpolating, they are not close to the immediate even neighbors, leading to a fractal like limiting function. In each step, the boxes represent the values in the even locations, the grey shaded circles correspond to the predicted values in the odds. In the next step of the refinement, boxes and circles together become boxes (evens).

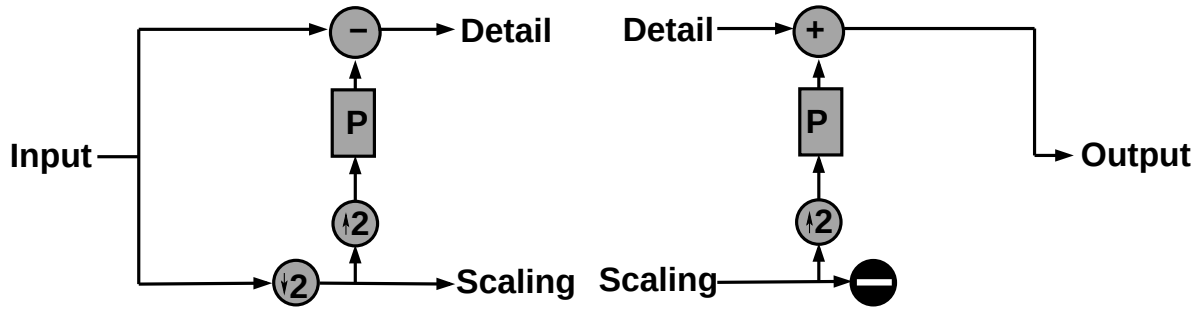

Figure 3: Lifted smoothing and an inverse transform. The inverse transform is not unique, because the decomposition is redundant.

In order to overcome the discontinuity upon subdivision, the even indexed input has to undergo the same smoothing operation. As a consequence, the prediction has to be evaluated on both odds and evens before being fed to all the fine scale coefficients. The detail branch of the lifting scheme in Figure 3 thus contains all fine scale indices, while the coarse scale branch keeps only the even indices from the fine scale set.

The scheme can be translated into the following algorithmic steps. **Given** the observations $Y_i = f(x_i) + \varepsilon_i$, **initialize** the lifted smoothing transform as

$$\begin{aligned} \mathcal{J}_J &= \{1, \ldots, n\} \text{ (index set)} \\ s_{J,k} &= Y_k \\ x_{J,k} &= x_k \end{aligned}$$

The initialization is identical to that of the classical, critically downsampled lifting scheme. Next, **iterate**:

- **Split** becomes **Copy**. All values are declared odds and half of them are copied into the even set. In the flow chart, this is represented as a full copy followed by a subsampling $(\downarrow 2)$ on the even branch. Formally, we have

$$\begin{aligned} \mathcal{J}_j = e &= \{2k | k \in \mathcal{J}_{j+1}\} \\ o &= \mathcal{J}_{j+1}. \end{aligned}$$

  The contrast with critically downsampled lifting lies in the definition of the odds, which is $o = \mathcal{J}_{j+1} \backslash \mathcal{J}_j$.

  The number of detail coefficients at each scale is approximately twice the number in a critically downsampled (fast) wavelet decomposition. Hence, the entire transform is overcomplete by a factor 2. This redundancy is smaller than the $\log(n)$ factor in a nondecimated translation-invariant wavelet transform. The overcompleteness is the price to reconciliate smoothness of reconstruction

and numerical stability.

- **Predict** the odds by the evens.

$$\boldsymbol{w}_j = \boldsymbol{s}_{j+1} - \mathrm{P}_j \cdot (\uparrow 2)\boldsymbol{s}_{j+1,e}.$$

  The symbol $(\uparrow 2)$ stands for upsampling. Strictly speaking, this is adding zeros between the elements of $\boldsymbol{s}_{j+1,e}$, so that the result has the same length as $\boldsymbol{s}_{j+1}$. The prediction matrix P_j is then a square matrix. In practice, the size expansion from $\boldsymbol{s}_{j+1,e}$ to $\boldsymbol{s}_{j+1}$ is of course incorporated into the prediction.

- Identify $\boldsymbol{s}_j = \boldsymbol{s}_{j+1,e}$.

As the transform is overcomplete, its inverse is not unique. The most straightforward reconstruction, depicted in Figure 2, simply inverts the prediction by $\boldsymbol{s}_{j+1} = \boldsymbol{w}_j + \mathrm{P}_j \cdot (\uparrow 2)\boldsymbol{s}_j$. This already delivers all fine scale coefficients on the odd branch.

3.2 Kernel smoothing prediction with pre-smoothing

Given that the reconstruction uses the even branch for the prediction only, exactly the same reconstruction still holds if the even branch is prefiltered before being used for prediction in the forward phase. Smoothing before subsampling reduces the aliasing, numerical, or biasing effects of simple subsampling. The resulting forward scheme is depicted in Figure 4, and corresponds to the following algorithm.

- **Copy** the full input on both even and odd branch . That is,

$$\begin{aligned} e &= \mathcal{J}_{j+1} \\ o &= e. \end{aligned}$$

- **Pre-smooth** the scaling coefficients, *then* keep the evens, i.e., subsample. This subsampling replaces the split in the version without pre-smoothing. The pre-smoothing or pre-filtering takes the form

$$\boldsymbol{s}_j = (\downarrow 2)\widetilde{\mathrm{H}}_j \cdot \boldsymbol{s}_{j+1}.$$

- **Predict**, as before, but now the prediction is constructed on a set of pre-smoothed values.

The variant with pre-smoothing can also be seen as a nonequidistant sample version of Burt and Adelson's Laplacian pyramid [2]. In the literature on this pyramidal scheme, the operation $(\downarrow 2)\widetilde{\mathrm{H}}_j$ is often termed *reduce*, while the operation $\mathrm{P}_j(\uparrow 2)$ is known as *expand*.

Since the inverse transform of 3 can still be used, independently from the design of the pre-smoother $\widetilde{\mathrm{H}}_j$, that pre-smoother can easily be made non-linear or data-adaptive.

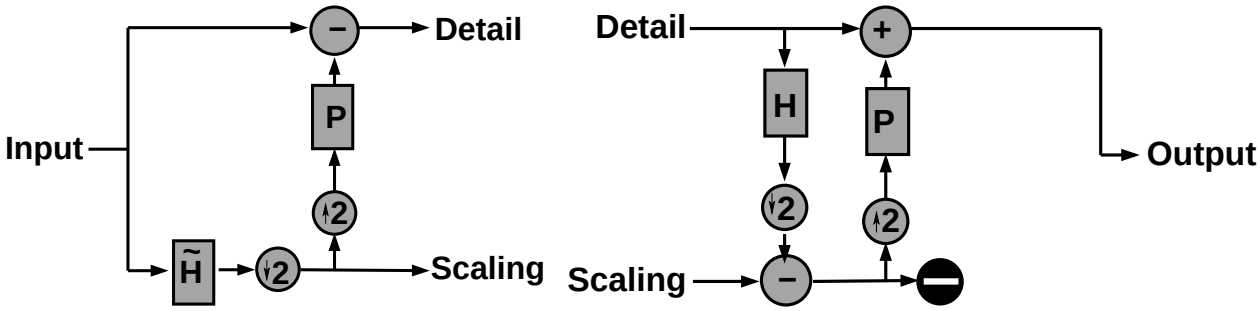

Figure 4: Lifted smoothing with pre-filtering on the evens. This can also be seen as a nonequidistant Laplacian pyramid. The inverse transform is not unique. The option $\mathrm{H}_j = 0$, corresponding to the scheme in Figure 3, is still one of the solutions that lead to perfect reconstruction, independent from $\widetilde{\mathrm{H}}_j$.

The inverse transform is, however, not unique, and in a more general form an additional filter H_j may operate in the reconstruction from details to scaling coefficients. Those details were found by

$$\boldsymbol{w}_j = \left(\mathrm{I} - \mathrm{P}_j(\uparrow 2)(\downarrow 2)\widetilde{\mathrm{H}}_j\right) \boldsymbol{s}_{j+1}. \tag{14}$$

On the other hand, simple inversion of the prediction in analysis leads to $\boldsymbol{s}_{j+1} = \boldsymbol{w}_j + \mathrm{P}_j(\uparrow 2)\boldsymbol{s}_j$. This expression provides the argument for the simple inverse transform in Figure 3, but it also imposes a condition on H_j in Figure 4. Indeed, in the scheme of Figure 4 we have

$$\boldsymbol{s}_{j+1} = (\mathrm{I} + \mathrm{P}_j(\uparrow 2)(\downarrow 2)\mathrm{H}_j)\, \boldsymbol{w}_j + \mathrm{P}_j(\uparrow 2)\boldsymbol{s}_j = (\boldsymbol{w}_j + \mathrm{P}_j(\uparrow 2)\boldsymbol{s}_j) + \mathrm{P}_j(\uparrow 2)(\downarrow 2)\mathrm{H}_j\boldsymbol{w}_j.$$

Filling in the simple inversion into the first half of this expression, we find that, for any $\boldsymbol{s}_{j+1}$,

$$\mathrm{P}_j(\uparrow 2)(\downarrow 2)\mathrm{H}_j\boldsymbol{w}_j = 0,$$

where $\boldsymbol{w}_j$ results from (14). This implies the following perfect reconstruction condition, written in three equivalent forms

$$\begin{aligned}
\mathrm{P}_j(\uparrow 2)(\downarrow 2)\mathrm{H}_j \left(\mathrm{I} - \mathrm{P}_j(\uparrow 2)(\downarrow 2)\widetilde{\mathrm{H}}_j\right) &= 0 \\
\left(\mathrm{I} - \mathrm{P}_j(\uparrow 2)(\downarrow 2)\mathrm{H}_j\right)\left(\mathrm{I} - \mathrm{P}_j(\uparrow 2)(\downarrow 2)\widetilde{\mathrm{H}}_j\right) &= \mathrm{I} - \mathrm{P}_j(\uparrow 2)(\downarrow 2)\widetilde{\mathrm{H}}_j. \\
\left(\mathrm{P}_j(\uparrow 2)(\downarrow 2)\mathrm{H}_j\right)\left(\mathrm{P}_j(\uparrow 2)(\downarrow 2)\widetilde{\mathrm{H}}_j\right) &= \mathrm{P}_j(\uparrow 2)(\downarrow 2)\mathrm{H}_j
\end{aligned} \tag{15}$$

Assuming that the matrix $\mathrm{P}_j(\uparrow 2)$ has full rank, multiplication on the left with its left inverse and on the right with $\mathrm{P}_j(\uparrow 2)$, leads to

$$(\downarrow 2)\mathrm{H}_j\mathrm{P}_j(\uparrow 2) = \left((\downarrow 2)\mathrm{H}_j\mathrm{P}_j(\uparrow 2)\right)\left((\downarrow 2)\widetilde{\mathrm{H}}_j\mathrm{P}_j(\uparrow 2)\right).$$

The option $\mathrm{H}_j = \widetilde{\mathrm{H}}_j$ has been proposed and elaborated in [8]. The option $\mathrm{H}_j = 0$ is a trivial solution. For any options H_j and $\widetilde{\mathrm{H}}_j$ with full rank reduce matrices, the condition above can be read as matrix $(\downarrow 2)\widetilde{\mathrm{H}}_j\mathrm{P}_j(\uparrow 2)$ having eigenvalue 1, with geometrical multiplicity equal to its size. This implies the condition that

$$(\downarrow 2)\widetilde{\mathrm{H}}_j\mathrm{P}_j(\uparrow 2) = \mathrm{I}. \tag{16}$$

For a full ranked H_j to exist in the synthesis, the presmoothing and prediction in the analysis thus needs to be biorthogonal.

3.3 Design of biorthogonal filters P_j, $\widetilde{\mathrm{H}}_j$ and H_j

The discussion in Section (3.2) has lead to two expressions, (15) and (16), for **perfect reconstruction** in case the synthesis uses a presmoothing H_j. In absence of pre-smoothing at the synthesis, perfect reconstruction is guaranteed by the lifting scheme itself. No additional condition is necessary in that case.

Nevertheless, it makes sense to impose (16), even if $\mathrm{H}_j = 0$, or at least a weaker form of that condition. Expression (16) can be interpreted as conditions for $\widetilde{\mathrm{H}}_j$, given a choice of P_j. Before motivating the use of (16) with arguments other than perfect reconstruction involving a filter H_j, it is necessary to elaborate on the prediction operator P_j. The prediction operator is the only one that plays a role in the subdivision scheme. It thus determines the **smoothness of the subdivision limit**, and related to that, the class of functions that can be produced by subdivision, i.e., the functions that can be represented with all detail coefficients zero. For good smoothness properties, it is important that all constant and linear functions are among that class [5]. Kernel smoothing does not reproduce linear functions, local linear polynomials do. This condition can be understood as follows. A scheme aiming at smooth reconstructions on a given grid should be able to reproduce that grid, i.e., the identical observations $\boldsymbol{y} = \boldsymbol{t}$. Otherwise the structure of $\boldsymbol{t}$ will be reflected in the reconstruction of $\boldsymbol{y}$. This paper suggests to use local linear polynomials, rather than kernel smoothing.

Since the decomposition is overcomplete, it is not guaranteed that function that *can* be represented with all details zeros, actually *has* all its details zero in a decomposition. The value of the details in a decomposition depend on all operations used in that decomposition, including $\widetilde{\mathrm{H}}_j$, which plays no role in the analysis of the subdivision scheme. Imposing (16) is sufficient to guarantee that all functions that can be produced by subdivision have zero details in a decomposition. Indeed, given arbitrary $\boldsymbol{s}_j$, let $\boldsymbol{s}_{j+1} = \mathrm{P}_j(\uparrow 2)\boldsymbol{s}_j$, then $\boldsymbol{s}_{j+1}$ can be represented with zero detail coefficients at scale j. Those detail coefficients follow from (14). Plugging in (16) into (14) yields a zero identity for any $\boldsymbol{s}_j$. As elaborated in Section 3.5, the expression can be weakened by imposing the zero details only for polynomials, such that the polynomials produces by the subdivision correspond to dual vanishing moments in the decomposition.

The pre-filter in the reconstruction, H_j, can be used to provide the synthesis with **primal vanishing moments**. That is, if the decomposition has lead to detailed coefficients $\boldsymbol{w}_j$ at successive scales j, then

the synthesis reconstructs a function

$$f_J(t) = \Phi_L(t) \cdot \boldsymbol{s}_L + \sum_{j=L}^{J-1} \Psi_j(t) \cdot \boldsymbol{w}_j,$$

where

$$O_j^{(p)} = \int_{-\infty}^{\infty} \Psi_j^T(t) t^p dt = \mathbf{0},$$

for a designer's choice of p. Define the scaling moments

$$M_j^{(p)} = \int_{-\infty}^{\infty} \Phi_j^T(t) t^p dt.$$

The values $M_j^{(p)}$ follow from the subdivision scheme, hence from the choice of P_j. The wavelet equation can be found by plugging in the identity matrix I into the reconstruction diagram, which leads to

$$\Psi_j(t) = \Psi_j(t) \cdot I = \Phi_{j+1}(t) \cdot \Big(\mathrm{I} - \mathrm{P}_j(\uparrow 2)(\downarrow 2)\mathrm{H}_j\Big).$$

Imposing zero moments $O_j^{(p)} = \mathbf{0}$ then leads to the primal moment equations

$$M_j^{(p)^T} \Big(\mathrm{I} - \mathrm{P}_j(\uparrow 2)(\downarrow 2)\mathrm{H}_j\Big) = \mathbf{0}. \tag{17}$$

This equation can be used to design H_j in function of the prediction operator P_j. Afterwards, the perfect reconstruction condition (15) can be used to find corresponding operations $\widetilde{\mathrm{H}}_j$. Condition (15) can also be used to translate (17) into a condition for $\widetilde{\mathrm{H}}_j$. Indeed, the combination of (15) and (17) implies immediately that

$$M_j^{(p)^T} \Big(\mathrm{I} - \mathrm{P}_j(\uparrow 2)(\downarrow 2)\widetilde{\mathrm{H}}_j\Big) = \mathbf{0}. \tag{18}$$

This equation can be interpreted as follows. Obviously, $\widetilde{\mathrm{H}}_j$ has an effect on the wavelet coefficients, but not on the wavelet basis functions. If (18) is satisfied, the coefficients $\boldsymbol{w}_j$ are such that at each scale j,

$$\int_{-\infty}^{\infty} t^p \Psi_j(t) \boldsymbol{w}_j dt = 0, \tag{19}$$

independent from which synthesis is being used. This *global primal moment* condition ensures — thanks to the overcompleteness of the transform and whatever primal wavelet function will follow from the reconstruction — that the decomposition into detail functions

$$\Delta f_j(t) = \sum_k w_{j,k} \psi_{j,k}(t),$$

satisfies $\int_{-\infty}^{\infty} \Delta f_j(t) t^p = 0$, even if $\int_{-\infty}^{\infty} t^p \Psi_j(t) dt \neq \mathbf{0}$.

The global primal moment condition (19) of the decomposition was inherited from local primal moments at the reconstruction phase through the perfect reconstruction condition (15) in way similar to how (16) is the translation of the perfect reconstruction to the decomposition stage only. Expressions (19) and (16) can be used to design the decomposition *before* the synthesis (instead of vice versa). Even is the synthesis does not construct basis functions with vanishing moments, imposing the global vanishing moment condition provides numerical stability of the transform.

This paper thus proposes to first investigate the subdivision through P_j and then to complete the design of the analysis through $\widetilde{\mathrm{H}}_j$, independent from the synthesis. The synthesis follows in the third step of the design. In the analysis, the degrees of freedom left by the constraints of vanishing moments and perfect reconstruction, can be used to maximize the sparsity of the decomposition. This is the topic of the next section.

3.4 Solving the design equations for the decomposition

Sparsity of the decomposition is promoted by a good choice of prediction operator and by sparse matrices $\widetilde{\mathrm{H}}_j$.

Given the prediction operator P_j, the matrix $\widetilde{\mathrm{H}}_j$ is designed to satisfy the conditions (16) and (18). This may proceed in three steps:

1. Find a sparse solution $\widetilde{\mathrm{H}}_{j,1}$ for (16). Sparsity in $\widetilde{\mathrm{H}}_{j,1}$ is imposed row by row: for each row i of $\widetilde{\mathrm{H}}_{j,1}$, it is checked which rows j in the ith column of P_j contain nonzeros. A solution with as few nonzeros as possible around column j in row i of $\widetilde{\mathrm{H}}_{j,1}$ is then tried until a zero residual is obtained.

2. Find a sparse solution $\widetilde{\mathrm{H}}_{j,0}$ for the homogeneous version of (16), i.e., for the condition $(\downarrow 2)\widetilde{\mathrm{H}}_j\mathrm{P}_j(\uparrow 2) = 0$.
   The techniques for obtaining sparsity are the same as in the first step of the algorithm.

3. Find a matrix X such that $\widetilde{\mathrm{H}}_j = \widetilde{\mathrm{H}}_{j,1} + X\widetilde{\mathrm{H}}_{j,0}$ satisfies (18). Writing

$$\begin{aligned} A &= {M_j^{(p)}}^T \mathrm{P}_j(\uparrow 2) \\ B &= {M_j^{(p)}}^T \Big(\mathrm{I} - \mathrm{P}_j(\uparrow 2)(\downarrow 2)\widetilde{\mathrm{H}}_{j,1}\Big) \\ O &= (\downarrow 2)\widetilde{\mathrm{H}}_{j,0}, \end{aligned}$$

   Expression (18) reduces to $AXO = B$, which can be solved in two stages

   (a) First solve $UO = B$ for U. Given that $M_j^{(p)}$ is non-sparse and has a limited number of rows, matrix U will not be sparse. As a matter of fact, the set of equations is typically redundant.

   (b) Find a sparse solution for the underdetermined equation $AX = U$.

3.5 Shared invariance instead of biorthogonality

The biorthogonality condition (16) is quite restrictive. This condition is sufficient, but not necessary, to guarantee that all functions produced by subdivision are decomposed in the overcomplete transform with all details coefficients zero. It should be noted that this condition is not fulfilled in absence of a prefilter, i.e., if $\widetilde{\mathrm{H}}_j = \mathrm{I}$. In the context of subdivision with smoothing prediction, the price to pay for this biorthogonality is, however, high. Indeed, the prefilter has to anticipate for the smoothing by doing the opposite thing, that is, blowing up highly frequent oscillations. Such operation introduces undesired oscillations, leading to numerical instabilities.

As an alternative, zero details coefficients can be imposed only for those functions for which the subdivision has been designed to reproduce. In particular, the local polynomial prediction scheme is intended to preserve polynomials up to degree $\tilde{n}-1$. That is, given the subsampled observational grid $\boldsymbol{t}_{j+1}$, if $s_{j+1,i} = t_{j+1,i}^p$ with $p = 0, 1, \ldots, \tilde{n}-1$, then $\Big(\mathrm{I} - \mathrm{P}_j(\uparrow 2)(\downarrow 2)\Big)\boldsymbol{s}_{j+1} = \mathbf{0}$. For such input, we impose that $\Big(\mathrm{I} - \mathrm{P}_j(\uparrow 2)(\downarrow 2)\widetilde{\mathrm{H}}_j\Big)\boldsymbol{s}_{j+1} = \mathbf{0}$. Denoting by $\mathrm{T}_{j+1}^{(\tilde{n})} = \Big[\mathbf{1}\ \boldsymbol{t}_{j+1} \ldots \boldsymbol{t}_{j+1}^{\tilde{n}-1}\Big]$ (where $\boldsymbol{t}_{j+1}^p$ is the vector with elements $t_{j+1,i}^p$), this is realized if

$$\widetilde{\mathrm{H}}_j \mathrm{T}_{j+1}^{(\tilde{n})} = \mathrm{T}_{j+1}^{(\tilde{n})}. \tag{20}$$

That is, $\mathrm{T}_{j+1}^{(\tilde{n})}$ must be invariant under $\widetilde{\mathrm{H}}_j$. Condition (20) replaces the biorthogonality condition in (16). A straightforward construction of a sparse solution for (20) follows by realizing that P_j satisfies a similar invariance expression, namely

$$\mathrm{P}_j(\uparrow 2)(\downarrow 2)\mathrm{T}_{j+1}^{(\tilde{n})} = \mathrm{T}_{j+1}^{(\tilde{n})}.$$

The prefilter can thus be constructed from the same family as the prediction, for instance, local polynomial prediction may be preceded by local polynomial prefiltering of the same polynomial degree, with possibly different bandwidths. When prefilter and prediction share invariance properties, the detail coefficients are offsets between observations and values that have gone through two similar smoothing operations. Such double smoothing also appears in other studies, for instance in the context of bias reduction [10] or robust nonparametric regression [11]. The approach through a shared invariance property is essentially different from the biorthogonality approach, which imposes prefilters that are, in a certain sense, the inverse of the prediction, so that both operations annihilate each other, rather than sharing any property.

A prefilter constructed from the same family as the prediction may not satisfy the global moment condition in (18). Let $\widetilde{\mathrm{H}}_{j,1}$ be such a solution. As in Section 3.4, this solution is corrected by $\widetilde{\mathrm{H}}_j = \widetilde{\mathrm{H}}_{j,1} + X\widetilde{\mathrm{H}}_{j,0}$, where $\widetilde{\mathrm{H}}_{j,0}$ satisfies the homogeneous equation $\widetilde{\mathrm{H}}_{j,0}\mathrm{T}_{j+1}^{(\tilde{n})} = \mathbf{0}$. The null space defined by this expression is much larger than that in Section 3.4, allowing sparser solutions.

Prefilters with shared invariance properties with the prediction are thus faster in construction, with a sparser and numerically more stable result than biorthogonal prefilters. The price to pay is that there is no nontrivial biorthogonal prefilter possible in the reconstruction. Such a prefilter can be replaced

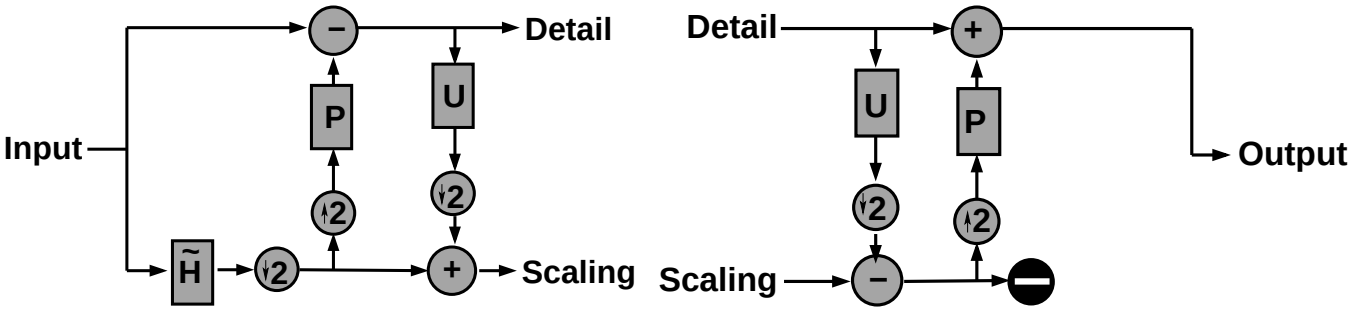

Figure 5: Lifted smoothing with pre-filtering on the evens. This can also be seen as a nonequidistant Laplacian pyramid. The inverse transform is not unique. The option $\mathrm{H}_j = 0$, corresponding to the scheme in Figure 3, is still one of the solutions that lead to perfect reconstruction, independent from $\widetilde{\mathrm{H}}_j$.

by an update step, as indicated in Figure 5 This update step can be designed for local primal vanishing moments, improving on the global primal vanishing moments by a prefilter in the analysis.

Remark

It can be observed that even without a perfect global vanishing moment, a prefilter has a stabilizing effect. A simple, straightforward prefiltering could, for instance, be a Haar prefilter $s_{j,k} = s_{j+1,2k-1} + s_{j+1,2k}$, or an observational grid dependent version of it (known as unbalanced Haar) $s_{j,k} = (\Delta_{j+1,2k-1} s_{j+1,2k-1} + \Delta_{j+1,2k} s_{j+1,2k})/\Delta_{j,k}$, where $\Delta_{j,k} = (x_{j,k+1} - 2x_{j,k} + x_{j,k-1})/2$. The explanation for the relatively good performance of such a simple scheme is that numerical stability follows from asymptotical global primal vanishing moments in any measure. This can be the Lebesgue measure, but also the empirical design measure [7]. Haar prefilters do not preserve lines on arbitrary observational grids, leading to a decomposition with more nonzeros than necessitated by the subdivision scheme.

Remark

If the reconstruction is designed to be independent from the prefilter, this prefilter can be made nonlinear or data-adaptive at no price.

3.6 Choosing the bandwidth and other transform parameters

The multiscale local polynomial transform presented in this paper depends on four parameters: the degree of the local polynomial, the kernel function, the bandwidth at each scale and the number of primal vanishing moments. The first three of these parameters together determine the limiting function of the subdivision process, hence the smoothness of the primal basis function.

The degree of the local polynomial directly defines the number of dual vanishing moments. That number equals the degree plus one. Vanishing moments control the sparsity of the representation.

Unlike in kernel or local polynomial smoothing methods, the kernel bandwidth is not in the first place a smoothing parameter, but rather one of the three parameters fixing the limiting function of the

subdivision process. The bandwidth has an important interpretation w.r.t. the scale in the multiresolution decomposition. As the interobservational distances are irregular, there is no unique scale following from the observations themselves, as would be the case in a classical wavelet analysis on equidistant data. The bandwidth takes the role of scale at each level. Associated to the role of the bandwidth is the property that the number of observations involved in the prediction filter may be time- or location-varying. This property compensates for possible differences in sample rates along the observational axis. It is reasonable to impose that the average number of observations involved in the prediction filters remains approximately the same across the consecutive levels during the transform. The bandwidths, i.e., the scale of the filtering, should thus be inversely proportional to the split or subsampling rate. In case of even-odd splitting, where roughly half of the observations are omitted at each stage, the bandwidth h_j at scale $j = L, \ldots, J-1$ can thus be taken to be $h_j = 2^{L-j} h_L$, where L is the coarsest scale (often set to be $L = 0$) and J is the resolution level of the observations. The latter equals $J = \log_2(N)$ in classical wavelet analysis. It is an interesting topic of further research to investigate if schemes with wider bandwidths at fine scales for additional smoothing are beneficial. If the bandwidth behaves as $h_j = q^{L-j} h_L$, with $1 < q < 2$, then the result after subdivision is smoother and still has finite support.

Next to the evolution of the bandwidth across scales is the choice of the finest scale bandwidth. Its value should not be too large as it is the only opportunity to analyze fine scale effects. On the other hand, precautions should be taken when bandwidths are small compared to the local sampling rate. Indeed, small bandwidths cover only a limited neighboring points. The number of neighbors may then be insufficient to comply with the number of vanishing moments imposed by the design of the transform. Secondly, those predicting points may happen to be all, or nearly all, on the same side of the predicted value. For local constant regression, i.e., kernel estimation, this poses no real problem, as the prediction is always convex. For local linear or higher order regression, the effect is large prediction coefficients, resulting in numerically unstable decompositions, especially if the prediction is followed by an update step for primal vanishing moments.

The software developed for this framework is equipped with a routine that allows "flexible" bandwidths so that narrow bandwidths can be extended in a way dependent on the local observational grid: if the global bandwidth for the current scale includes too few adjacent points for stable prediction, complying with the design conditions, then the algorithm searches for more neighbors left and right from the current band.

3.7 Convex prediction coefficients

Although instabilities have already been reduced using smoothing instead of interpolation and using flexible bandwidths, further improvement is necessary by imposing convex prediction coefficients. Convex coefficients follow automatically in the cases of linear interpolating prediction and constant local least squares prediction. In general local least squares prediction, and given a flexible bandwidth, as discussed above in Section 3.6, we impose that all prediction coefficients are positive. Since the first dual vanishing

moment requires them to sum up to one, the positivity condition implies convexity. Local polynomial regression using least squares with the positivity condition leads to a non-convex combinatorial optimization problem, namely, find $\widehat{\boldsymbol{\beta}}$ that minimizes

$$\|W_{j+1,e}(t)(\boldsymbol{Y} - \mathrm{T}_{j+1,e}\boldsymbol{\beta})\|$$

for $\boldsymbol{\beta}$ satisfying the conditions (for $i \in e$)

$$\frac{\partial(\mathrm{T}^{(\tilde{n})}(t)\boldsymbol{\beta})}{\partial Y_i} \geq 0.$$

The prediction $P_j(t; \boldsymbol{t}_{j+1,e}) = \mathrm{T}^{(\tilde{n})}(t)\widehat{\boldsymbol{\beta}}$ replaces Expression (13). The software written for this paper finds a local optimum for this problem. As the neighborhoods for the prediction of a given point are local, the complexity of the optimization problems remains under control.

4 Illustrations and simulations

4.1 The frame functions

As the proposed signal decomposition is redundant, the building blocks do not constitute a basis, but rather a frame. Figure 6 compares scaling functions from subdivision with interpolating polynomial predictions and scaling functions from subdivision with local polynomial smoothing as prediction step. For the interpolating subdivision, cubic polynomials were used. The resulting scaling function is smooth, corresponding to theoretical results [4], but, depending on the grid of sample locations, the function may show heavy side lobes. This is due to the fact that every prediction involves two neighbors on the left and two on the right. Such an approach does not weight the importance of the neighbors according to their distance. Using kernel (local constant) smoothing as prediction is insufficient to capture the structure of the grid. Indeed, as local constants cannot reproduce the function $y(t) = t$, it can be expected that the locations t_i have an effect on the decomposition.

4.2 The analysis or decomposition

The frame functions are determined by the reconstruction, and thus by the refinement or subdivision scheme. As the proposed representation is overcomplete, the decomposition into this frame is not unique. In particular, the prefilter step is subject of design. As discussed in Sections 3.3 and 3.5, biorthogonal design on irregular grids is time consuming and restrictive, leading to decompositions that are non-sparse, unstable, or both. But also for prefilters with shared invariance properties, the current implementation of the design methods in Sections 3.4 and 3.5 is not always successful in combining the properties (mostly dual vanishing moments) with sparse and stable preservation of the global primal vanishing moments. Since an update step has already been proposed as an alternative for a biorthogonal pair of prefilters and

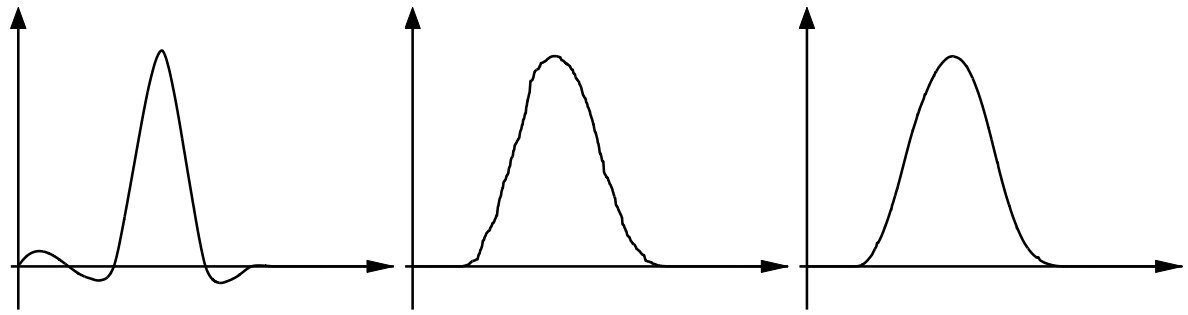

Figure 6: Scaling functions for (respectively) cubic interpolation prediction, local constant (kernel) smoothing prediction and local linear smoothing prediction.

prediction, this update step can also be used to overcome the inconveniences of prefilter that violates the global primal moment condition.

4.3 A denoising example

The following illustrations and simulations were performed on the test signal

$$f(t) = \begin{cases} \sin(9x) & \text{for} \quad 0 \le x < 0.3 \\ 4 & \quad\quad\; 0.3 \le x < 0.4 \\ 4(0.62 - x)^2/(0.62 - 0.4)^2 & \quad\quad\; 0.4 \le x < 0.62 \\ 8(1 - \sqrt{1 - (x - 0.62)^2/(0.7 - 0.62)^2}) & \quad\quad\; 0.62 \le x < 0.7 \\ 8(1 - \sqrt{1 - (0.85 - x)^2/(0.85 - 0.7)^2}) & \quad\quad\; 0.7 \le x < 0.85 \\ 2 + 9(x - 0.85) & \quad\quad\; 0.85 \le x \le 1 \end{cases}$$

The signal was sampled at $n = 2067$ locations t_i which were generated as (ordered) independent realizations of uniform random variables on $[0, 1]$. Additive normal (Gaussian) noise $\boldsymbol{\varepsilon}$ was added to this sample, to observe $\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\varepsilon}$ in a signal-to-noise ratio of

$$\mathrm{SNR} = 20 \log_{10} \left(\frac{\|\boldsymbol{f}\|}{\|\boldsymbol{\varepsilon}\|} \right) = 10\mathrm{dB}.$$

Figure 8 compares the noise reduction capacities of several options in a nonequispaced Laplacian pyramid scheme. All methods thus adopted the factor 2 redundancy of the pyramid, including the interpolating prediction schemes, were such redundancy is not strictly necessary. The choice for redundant interpolating prediction is motivated by fairness of comparison, as reconstruction from overcomplete schemes may result in additional smoothing. Experiments (not displayed in the figures) suggest that prefilters in an interpolating scheme have relatively little impact.

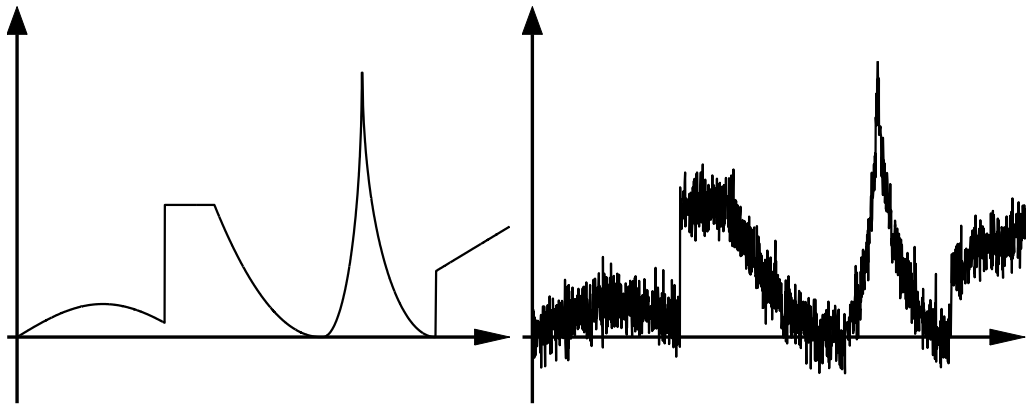

Figure 7: Test signal and noisy sample. Sample size is $n = 2067$.

The linear interpolating prediction with two-taps update of Figure 8(a), which the nonequispaced extension of the Cohen-Daubechies-Feauveau (CDF) wavelets with 2 primal and 2 dual vanishing moments, performs quite well in the sense of numerical stability: the estimation is fairly unbiased. Nevertheless, smoothness of this decomposition is limited, as it only reproduces straight lines. Trying to increase the smoothness, using cubic interpolating prediction, we arrive at 8(b). This decomposition is no longer an extension of a member of the CDF family. The result is smoother indeed, but peaks are less sharply reconstructed, and above all, the reconstruction shows unpleasant blobs, due to numerical instabilities. These blobs may be more serious in other settings than the one shown in the figure. Figures (c) and (d) compare local constant and local linear predictions. The local constant one clearly struggles with the irregularity of the point set. Both schemes show a remarkable and unacceptable vertical shift, due to the lack of local primal moments, i.e., building blocks with zero integral. This is remedied by adding an update step in Figures (e) and (f). Both Figures also introduce a local linear presmoothing. The difference between (e) and (f) is that in (f) the smoothing operations are constraint to have convex coefficients, which leads to a smoother and more stable result, with sharper reconstructions of the peaks.

Figure 9 summarizes the result of a simulation study of a hundred times 2067 noisy observations from the signal in Figure 7. The boxplot represents the observed difference in output signal-to-noise ratios of the two most stable routines in the previous discussion, namely convex local linear smoothing prediction with presmoothing on one hand and linear interpolating prediction on the other hand, both methods equipped with a two-taps update for local primal vanishing moments. These methods correspond, respectively, to the results displayed in Figures 8(f) and (a).

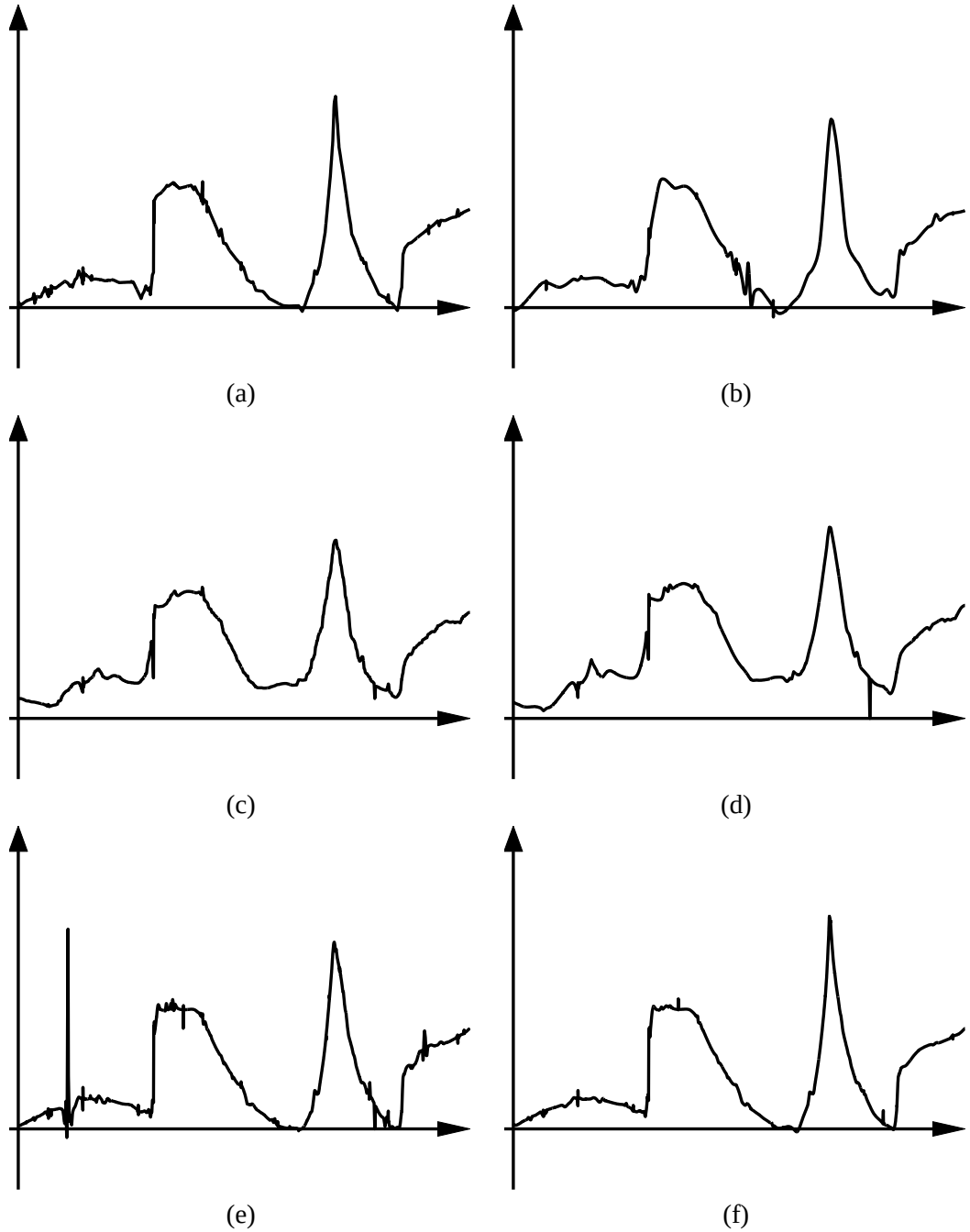

Figure 8: Denoising the observations from Figure 7. (a) Using linear interpolating prediction and two-taps update. (b) Using cubic interpolating prediction and two-taps update. (c) Using local constant smoothing prediction, no update, no prefilters. (d) Using local linear smoothing prediction, no update, no prefilters. (e) Using local linear smoothing prediction, two-taps update, linear smoothing prefilter. (f) Using convex local linear smoothing prediction, two-taps update, linear smoothing prefilter.

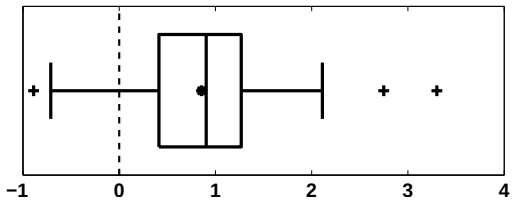

Figure 9: Boxplot of the difference in output SNR between convex local linear smoothing prediction with presmoothing and linear interpolating prediction, both methods with two-taps update. The population of differences has a positive mean, indicating that local linear smoothing is superior to linear interpolation as prediction method.

5 Concluding discussion

This paper has introduced the idea of multiscale smoothing, by plugging in well-known statistical methods of kernel smoothing and local polynomial smoothing into a lifting scheme. The decomposition can be applied to irregularly observed data, with a sample size not necessarily dyadic (a power of two). As explained in the paper, when smoothing replaces interpolation as basic lifting step, a slight redundancy in the transform is necessary to deal with problems of discontinuity. This overcompleteness is of factor 2, which is far below the redundancy factor in a classical non-decimated wavelet transform. The scheme can be seen as a nonequispaced version of a Laplacian pyramid. The paper discusses an extension of the prefiltered analysis and reconstruction in such a scheme, thereby illustrating that some options in the regular sample case proposed in [8] become problematic in the irregular sample case.

Using smoothing as basic operation in the subdivision or refinement process behind a wavelet transform offers several advantages and perspectives.

1. Compared to interpolation as basic design tool, smoothing allows easier control on the numerical condition of the refinement process. No interpolation beyond simple polylines performs with good numerical condition on irregular grids. Next to interpolation in a lifting scheme, wavelets on irregular point sets could also be constructed by elaboration from existing two-scale equations in such general settings. Examples include B-splines [5]. As far as experiments allowed to conclude, these constructions equally suffer from numerical problems on irregular point sets. The reason could be called a “mixture of scales”, meaning that irregular point sets typically show an intermittent density of observations, which conflicts with the basic idea of a multiscale analysis, where each level is characterized by its scale of operations.

2. Smoothing as basic operation suffers less from intermittent densities of observations, but it requires redundancy for the sake of smooth reconstructions. It could be conjectured that the combination of smooth and stable basis functions in a critically downsampled wavelet transform on a regular dyadic grid is an exceptional coincidence. Orthogonal (thus stable) constructions on irregular data

structures exist, such as on graphical data [16], but these do not consider any sort of smoothness on these graphs. The combination of smoothness and stability seems to require overcompleteness.

3. Using smoothing as basic operation in subdivision has another advantage compared to interpolating refinement. The output of the smoothing operation can be used in a refinable measure of local smoothness. In particular, as local linear polynomial smoothing allows to estimate the local derivative of the underlying noise-free function, a parallel multiscale analysis for the derivatives can be constructed for using in estimation and hypothesis testing.

References

[1] A. Antoniadis and J. Fan. Regularized wavelet approximations. *J. Amer. Statist. Assoc.*, 96(455):939–955, September 2001.

[2] P. J. Burt and E. H. Adelson. Laplacian pyramid as a compact image code. *IEEE Trans. Commun.*, 31(4):532–540, 1983.

[3] T. Cai and L.D. Brown. Wavelet shrinkage for nonequispaced samples. *Annals of Statistics*, 26(5):1783–1799, 1998.

[4] I. Daubechies, I. Guskov, and W. Sweldens. Regularity of irregular subdivision. *Constructive Approximation*, 15(3):381–426, 1999.

[5] I. Daubechies, I. Guskov, and W. Sweldens. Commutation for irregular subdivision. *Constructive Approximation*, 17(4):479–514, 2001.

[6] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4(3):245–267, 1998.

[7] V. Delouille, J. Simoens, and R. von Sachs. Smooth design-adapted wavelets for nonparametric stochastic regression. *J. Amer. Statist. Assoc.*, pages 643–658, 2004.

[8] M. N. Do and M. Vetterli. Framing pyramids. *IEEE Transactions on Signal Processing*, 51(9):2329–2342, 2003.

[9] M. Flierl and P. Vandergheynst. An improved pyramid for spatially scalable video coding. In *Proc. IEEE Int. Conf. on Image Proc. — ICIP '05*, pages 878–881, 2006.

[10] H. He and L.-S. Huang. Double smoothing for bias reduction in local linear regression. *Journal of Statistical Planning and Inference*, 139:1056–1072, 2009.

[11] R.-Ch. Hwang, Z.-H. Lin, and C.K. Chu. Double smoothing robust estimators in nonparametric regression. *Sankhya: The Indian Journal of Statistics*, 71(A, Part 2):298–330, 2009.

[12] M. Jansen and P. Oonincx. *Second generation wavelets and applications*. Springer, 2005.

[13] A. Kovac and B. W. Silverman. Extending the scope of wavelet regression methods by coefficient-dependent thresholding. *J. Amer. Statist. Assoc.*, 95:172–183, 2000.

[14] L. Liu, L. Gan, and T.D. Tran. Lifting-based laplacian pyramid reconstruction schemes. In *Proc. IEEE Int. Conf. on Image Proc. — ICIP '08*, pages 2812–2815, 2008.

[15] R. Lorentz and P. Oswald. Criteria for hierarchical bases in sobolev spaces. *Appl. Comp. Harmon. Anal.*, 8(1):32–85, 2000.

[16] S. K. Narang and A. Ortega. Perfect reconstruction two-channel wavelet filter-banks for graph structured data. To Appear:To Appear, 2012.

[17] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comp. Harmon. Anal.*, 3(2):186–200, 1996.

[18] W. Sweldens. The lifting scheme: a construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546, 1998.

[19] W. Van Aerschot, M. Jansen, and A. Bultheel. Adaptive splitting for stabilizing 1-d wavelet decompositions. *Signal Processing*, 86(9):2447–2463, 2006.

[20] E. Vanraes, M. Jansen, and A. Bultheel. Stabilizing wavelet transforms for non-equispaced data smoothing. *Signal Processing*, 82(12):1979–1990, December 2002.