MULTISCALE KERNEL SMOOTHING USING A LIFTING SCHEME

Maarten Jansen[1], *and Christophe Damerval*[2]

[1]Dept. of Mathematics, K.U.Leuven
Celestijnenlaan 200-B, B-3001, Leuven, Belgium
phone: ++32 (0)16 32 27 87, email: maarten.jansen@wis.kuleuven.be
web: http://perswww.kuleuven.be/maarten_jansen
[2]Dept. of Computer Science, K.U.Leuven
Celestijnenlaan 200-A, B-3001, Leuven, Belgium
phone: ++32 (0)16 32 70 81, email: christophe.damerval@cs.kuleuven.be

ABSTRACT

This paper discusses the idea of a lifting scheme for multiscale implementation of kernel estimation procedures used in statistical estimation. The resulting decomposition is related to the Burt-Adelson pyramid, but the design of the filters is well adapted to nonequispaced samples. The proposed decomposition has an oversampling rate of 2, where the oversampling can be seen as an alternative to primal lifting steps (update steps) as a tool for stabilising and anti-aliasing. We then propose an adaptive version of this multiscale kernel estimation with truncated kernels. Truncated kernels allow sharp representations of jumps. Illustrations show that our method is numerically well conditioned, suffers less from visual effects due to false detections, and allows indeed sharp transitions if equiped with an adaptive choice among truncated kernels. All variants of the proposed method have linear computational complexity.

Key words: wavelet; lifting; kernel; adaptive; smoothing; thresholding

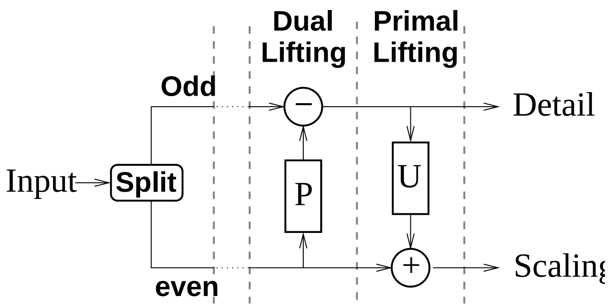

Figure 1: A general lifting scheme is a (mostly alternating) sequence of primal and dual lifting steps, initiated by a splitting stage. The dots between the splitting stage and the dual lifting step in the diagram indicate that a general lifting scheme may consist of more than one sequence of dual and primal lifting steps: after a primal step, a new dual step may follow.

1. INTRODUCTION

The lifting scheme [20, 21, 22] is an implementation of a filterbank in a wavelet transform. A wavelet filterbank is one stage in the multiscale decomposition that transforms scaling coefficients $s_{j+1,k}, k = 1, \ldots, 2^{j+1}$ at fine scale $j+1$ into scaling coefficients $s_{j,k}, k = 1, \ldots, 2^j$ at coarse scale j plus detail or wavelet coefficients $w_{j,k}$ at scale j. Scaling coefficients are further processed in the filterbank of the next stage. The lifting implementation is a sequence of lifting steps, as indicated in Figure 1. Lifting steps come in two main types: update or primal lifting steps and prediction or dual lifting steps. Prediction steps compute the offset of a subset of the input samples from a prediction (low pass filter) based on the complementary subset of input samples. The resulting operation can be seen as a high pass filter on the input samples. The update steps operate as low pass filter on the subset of input values that proceeds to the next, coarse scale. It can be seen as an anti-aliasing operation after subsampling the input stream. All classical wavelet decompositions can be reorganised as a sequence of lifting steps. The lifting implementation offers interesting benefits. First, the number of computations is lower than in the classical filterbank [15] implementation or in the polyphase implementation [19] (of which the lifting scheme is a further elaboration). Second, the inverse transform follows immediately, since every single lifting step can easily be undone. This is because filter operations in the lifting scheme take place on copies of input, while the original input of the current step proceeds untouched on another branch to the next step. As a consequence, the input of the filter step is still available after the filtering has taken place. Besides the algorithmic benefits, the lifting scheme also introduced an important conceptual novelty: the scheme serves as a general framework for the design of new types of wavelet transforms. First, the filter operations in a lifting scheme can easily be made nonlinear, and even data adaptive [3, 4, 17, 18, 16, 14, 12]. The most well known example of a nonlinear lifting scheme is the wavelet transforms that maps integers onto integers [2]. Second, the data that are analysed need not be sampled on equidistant intervals [6, 7, 5, 13, 9, 8, 24, 23]. This paper discusses a multiscale version of kernel (density) estimation where the samples may be irregular.

2. MULTISCALE KERNEL AND LOCAL POLYNOMIALS

Kernel estimation is a well-known technique in nonparametric statistics for regression of smooth functions. The noise can be additive normal, but also multiplicative, Poisson distributed. An important application is probability density estimation from observations.

Given n observations (x_i, y_i), a kernel estimator can be defined in every point x as

$$\hat{f}(x) = \frac{\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right) y_i}{\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right)} \quad (1)$$

In this expression $K(u)$ is a kernel function, typically a function with bounded support and $\int_{-\infty}^{\infty} K(u)du = 1$. The factor h is the bandwidth. Optimal choice of the bandwidth (i.e., the bandwidth that leads to the output with smallest error compared the true signal, i.e., the output with maximum signal-to-noise ratio), or estimation of the optimal value, is an important topic in Kernel smoothing. Obviously, the bandwidth can be seen as a scale parameter. In a multiscale version, the bandwidth will have several values h_j, where j is the index refering to scale (resolution level).

Our proposed multiscale decomposition uses expression (1) as a prediction operator in a lifting scheme, i.e.,

$$P(x; \mathbf{x}_{j+1,e}, \mathbf{s}_{j+1,e}) = \frac{\sum_{k=1}^{2^j} K\left(\frac{x-x_{j+1,2k}}{h_{j+1}}\right) s_{j+1,2k}}{\sum_{k=1}^{2^j} K\left(\frac{x-x_{j+1,2k}}{h_{j+1}}\right)} \quad (2)$$

This is the prediction operator, based on the even indexed locations and scaling coefficients $\mathbf{x}_{j+1,e}$ and $\mathbf{s}_{j+1,e}$, at scale $j+1$, evaluated in a point x. Note that, for sake of invertibility, the kernel smoothing takes place on the even indexed samples only. A naive approach would be to plug in (2) as prediction step into the scheme of Figure 1 and find some appropriate update step. That is, the wavelet coefficients at scale j equal $w_{j,k} = s_{j,k} - P(x_{j+1,2k+1}; \mathbf{x}_{j+1,e}, \mathbf{s}_{j+1,e})$. This is problematic for the following reason. The prediction value in an odd indexed point $x_{j,2k+1}$ at scale j depends on all even indexed samples within bandwidth h_{j+1} distance. The prediction operator already includes a smoothing. This is in contrast to, for instance, polynomial or average polynomial predictions [22]. As a consequence, if $x_{j,2k+1}$ is close to one of its even neighbours, say $x_{j,2k}$, the prediction value is not close to the observation in that even value. In other words

$$\lim_{u \to x_{j,2k}} P(u; \mathbf{x}_{j+1,e}, \mathbf{s}_{j+1,e}) \neq s_{j,2k}$$

Polynomial prediction (i.e., the Deslauriers-Dubuc [10] refinement scheme) has this continuity property. In absence of this continuity, the limiting function of a subdivision scheme (i.e., the inverse transform on an infintely fine grid with all detail coefficients equal to zero) cannot possibly be smooth, and hence the scheme is of no practical use for applications as smoothing or compression.

In order to make the output of the refinement (subdivision) step continuous, the even indexed observations should be filtered as well. The classical update lifting step would not be of any help here as it would not have any effect on the subdivision process. Instead we apply the same smoothing on the even indexed coefficients as well. Because this filtering step would not be invertible as such, we need to store the difference between input and output. The result is the scheme in Figure 2. Filter P_e is the kernel smoothing evaluated at the even indexed locations, while P_o is the same kernel smoothing in the odd locations. The presented scheme computes offsets (details) for both even and odd coefficients, such that the number of detail coefficients equals the length of the input and (up to boundary effects) the overall transform

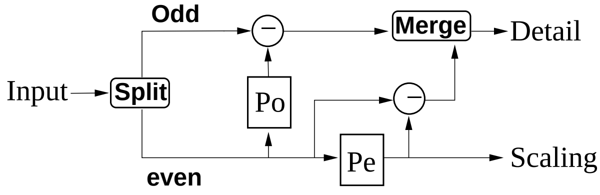

Figure 2: A lifting scheme with predictions on both even and odd indexed samples. Although the operations *Pe* and *Po* can be designed separately, practical implementations use the same operations, evaluated in the even and odd indexed locations respectively, for reasons of continuity.

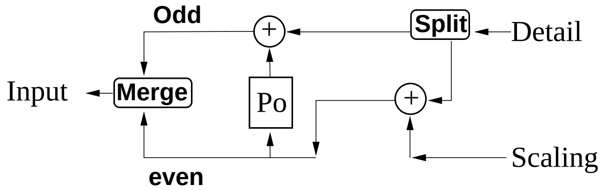

Figure 3: Inverse lifting scheme for Figure 2

doubles the number of data. The kernel smoothing prediction contains a parameter h_j, which is the kernel bandwidth, mentioned before. The optimal bandwidth is generally lower than what one would expect in a traditional kernel smoothing routine. The heuristical choice adopted in Section 4 has been found to approach the optimal choice quite well in the given simulation settings.

The inverse transform first reconstructs the even indexed observations and then uses them as input for the kernel smoothing procedure. As indicated in Figure 3, there is no need to apply the kernel smoothing onto the even indexed locations, as we already have the smoothed and original values at that moment. The scheme as a whole satisfies the perfect reconstruction property.

Besides smoothness of subdivision, the oversampling can also replace the use of an update step.

Our scheme is related to the well known class of Burt-Adelson pyramids [1]. The blurring in our scheme takes place on the even indexed observations and is used as prediction on the odd indexed observations. The scheme is perfectly adapted to data on irregular point sets: the filters and even the filter lengths (number of nonzeros) depend on the location of the neighbouring points. This is in contrast to the Deslauriers-Dubuc scheme where the number of nonzeros is fixed. A fixed number of taps may lead to instabilities if neighbouring points are at highly nonequidistant intervals, thereby mixing up different scales within a single resolution level [24, 23].

The kernel smoothing procedure can be seen as a running weighted average of neighbouring observations. If all observations y_i have the same, constant value c, then so have all predictions, and hence all detail coefficients at all levels will be zero. In other words, the scheme has one dual vanishing moment. In order to enhance the number of dual vanishing moments (thereby creating more sparsity and better compressibility), one could replace the kernel smoothing by a more advanced local polynomial smoothing.

Finally, we also note that since kernel methods are useful in settings beyond the classical additive normal noise case, our decomposition is expected to be promising in — for instance — multiplicative (Poisson) noise reduction.

3. ADAPTIVE LIFTING

Adaptive lifting for denoising [16, 14] is less restricted by side conditions than adaptive lifting schemes for application in data compression. Indeed, no special attention needs to be paid to the compressibility of the adaptivity information itself. This side information can be stored and used upon reconstruction. Just as in [16, 14], we propose an adaptive scheme that choses among several prediction schemes, based on the abolute values of the resulting coefficients. That is, let

$$P^{(i)}(x;\mathbf{x}_{j+1,e},\mathbf{s}_{j+1,e}) = \frac{\sum_{k=1}^{2^j} K_i\left(\frac{x-x_{j+1,2k}}{h_{j+1}}\right) s_{j+1,2k}}{\sum_{k=1}^{2^j} K_i\left(\frac{x-x_{j+1,2k}}{h_{j+1}}\right)}$$

with different values of i be a collection of prediction schemes, and define the resulting candidate wavelet coefficients as

$$w_{j,k}^{(i)} = s_{j+1,2k+1} - P^{(i)}(x_{j+1,2k+1}).$$

Then we could pick the final coefficient as

$$w_{j,k} = w_{j,k}^{(i^*)} \text{ where } i^* = \arg\min_i \left|w_{j,k}^{(i)}\right|. \tag{3}$$

Inspired by an edge adaptive method in (one-scale) kernel smoothing [11], we choose among three kernels K_i. First, let K_0 be the kernel used in Equation (2). Then, we define $K_R(x)$ as the right-truncated version of $K_0(x)$ and $K_L(x)$ the left-truncated version. That is, $K_L(x) = K_0(x) \cdot I(x \geq 0)$ where $I(x \geq 0)$ is the Heaviside step function (or indicator function or characteristic function on the positive axis). We let $i \in \{0, L, R\}$.

The selection of the final coefficient is a bit different from (3), as proposed in [16, 14]. Indeed, suppose that all three candidates would yield a noise-free value of zero (or close to zero). Such a situation is far from unlikely, as smooth intervals in classical wavelet analysis lead to negliglible coefficients. The eventual selection in (3) would then heavily depend on the noise. As illustrated in the simulation section, our experiments seem to indicate that this results in small wiggly effects in the reconstruction and especially in a tendency towards the detection of false jumps in otherwise smooth but non-constant functions. In order to reduce these random effects, the truncated kernels are only taken into account if they deliver a coefficient which is (say) 3 times smaller in magnitude than the full kernel's coefficient, that means if the full kernel's coefficient is *significantly* higher (and hence less favourable) than one of the truncated alternatives.

4. ILLUSTRATIONS, SIMULATIONS AND DISCUSSION

We illustrate our methods with the piecewise smooth "skyline" signal, depicted in Figure 4 and defined for $x \in [0,1]$

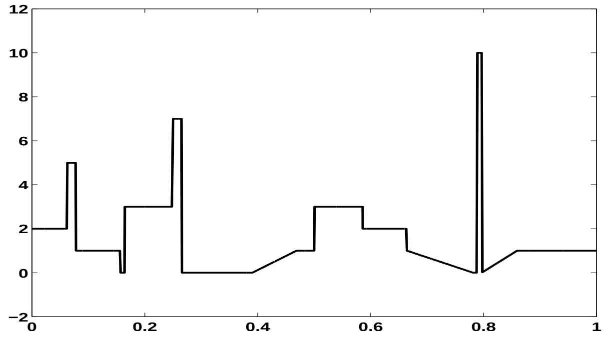

Figure 4: The "skyline" testfunction. This signal combines jumps, constant and linear intervals.

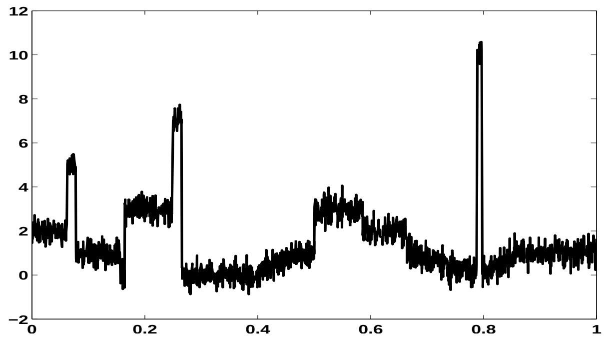

Figure 5: 2049 noisy observations of the signal in Figure 4. The locations of these observations are drawn from a uniform distribution on $[0,1]$.

as

$$f(x) = \begin{cases} 2 & \text{if } x \leq 1/16 \\ 5 & \text{if } 1/16 < x \leq 5/64 \\ 1 & \text{if } 5/64 < x \leq 5/32 \\ 0 & \text{if } 5/32 < x \leq 21/128 \\ 3 & \text{if } 21/128 < x \leq 1/4 \\ 7 & \text{if } 1/4 < x \leq 17/64 \\ 0 & \text{if } 17/64 < x \leq 25/64 \\ \frac{64}{5}x - 5 & \text{if } 25/64 < x \leq 15/32 \\ 1 & \text{if } 15/32 < x \leq 1/2 \\ 3 & \text{if } 1/2 < x \leq 75/128 \\ 2 & \text{if } 75/128 < x \leq 85/128 \\ \frac{20}{3} - \frac{128}{15}x & \text{if } 85/128 < x \leq 25/32 \\ 0 & \text{if } 25/32 < x \leq 1615/2048 \\ 10 & \text{if } 1615/2048 < x \leq 51/64 \\ 16x - \frac{51}{4} & \text{if } 51/64 < x \leq 55/64 \\ 1 & \text{if } 55/64 < x \leq 1 \end{cases}$$

The simulation is set up as follows. We first generate and order $n = 2049$ data points x_i from a uniform distribution on $[0,1]$. Then we generate noisy observations $Y_i = f(x_i) + \eta_i$, where $\eta_i \sim N(0,\sigma^2)$ and $\sigma = 1/3$. The observations are shown in Figure 5.

We apply an adaptive multiscale kernel smoothing procedure with cosine kernels, i.e., $K(x) = I(|x| < 1) \cdot \cos(\pi x/2)$ and bandwidths $h_J = 3(x_n - x_1)/n$ (such that on average the

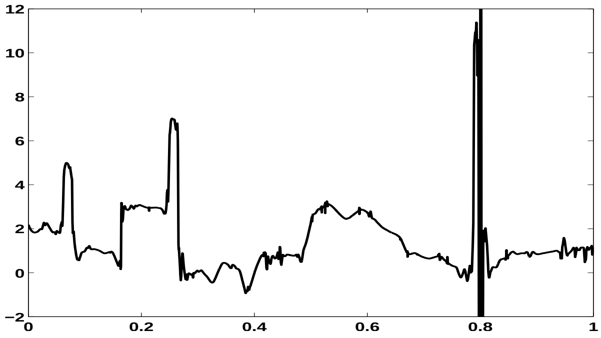

Figure 6: Output from classical lifting with a cubic interpolating polynomial as prediction step. The result is heavily biased due to numerical instability, specific for non-equidistant grids.

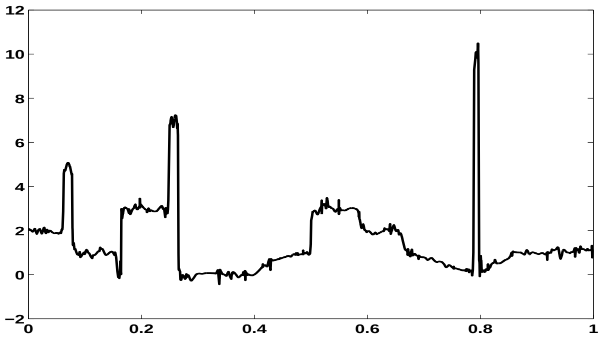

Figure 7: Output from lifting on a grid adaptive coarsening (splitting) routine. The grid adaptive coarsening eliminates most numerical problems.

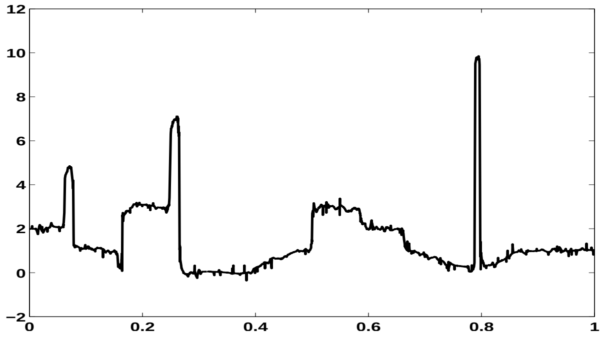

Figure 8: Output from a multiscale kernel transform. No numerical problems here, even with a simple even-odd splitting.

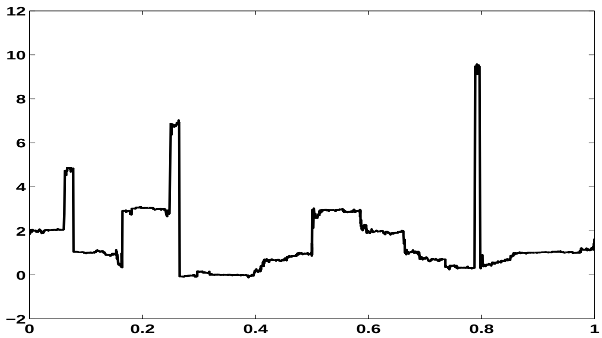

Figure 9: Output from an adaptive multiscale kernel transform. Each coefficient results from selection among a two sided kernel prediction and two one sided kernel predictions. Reconstructions of singularities are sharp, but the method sometimes detects jumps that are in fact gradual transitions (slopes).

fine scale kernels contain three data points: left, central, right) and $h_j = 2h_{j+1}$. Well founded choices for these parameters are of course subject of further research.

In order to evaluate our method, we compare it with related lifting schemes on for irregular point sets. As a lifting scheme we use a cubic interpolating polynomial as prediction step followed by a two taps update, designed such that the primal wavelet basis has two vanishing moments [22]. The output of a level-dependent minimum mean squared error threshold appears in Figure 6. The figure shows an unacceptable bias due to bad numerical condition of classical lifting. The numerical problems are specific for inhomogeneous grids and have been reported in [24, 23]. It should be emphasized that these problems occur even if the grid is relatively homogeneous as in this case: points were generated uniformly on the interval.

Figure 7 shows the output of the same lifting scheme applied on a grid adaptive coarsening procedure, as elaborated in [24], again with a level-dependent minimum mean squared error threshold approach. The numerical problems have been solved, but the reconstruction shows lots of prominent effects from false positives, i.e., coefficients that falsely survived the threshold.

Next three figures illustrate the methods presented in this text. First, Figure 8 plots the reconstruction from a plain multiscale kernel transform followed by a level-dependent minimum mean squared error threshold. Although the coarsening proceeds by even-odd splitting, no numerical problems occur, thanks to the fact that the prediction with kernels yields prediction coefficients that are always bounded between 0 and 1.

Figure 9 contains the output from the adaptive multiscale kernel transform with one full and two truncated kernels and decision rule (3). Jumps are much sharper in this reconstruction. The method has a tendency, however, to reconstruct the linear sections as a sequence of jumps as well. A better compromise is probably Figure 10, where a truncated kernel is used as prediction only if it delivers a *signicantly* smaller coefficient than the two-sided kernel.

5. CONCLUSIONS, ONGOING AND FUTURE RESEARCH

We have introduced a multiscale version of kernel smoothing, using a lifting scheme construction. We have defined forward and inverse transforms and proposed an adaptive version of the scheme, driven by a statistical hypothesis testing procedure. Issues under current investigation include a proper choice of the bandwidth at each scale. We also in-

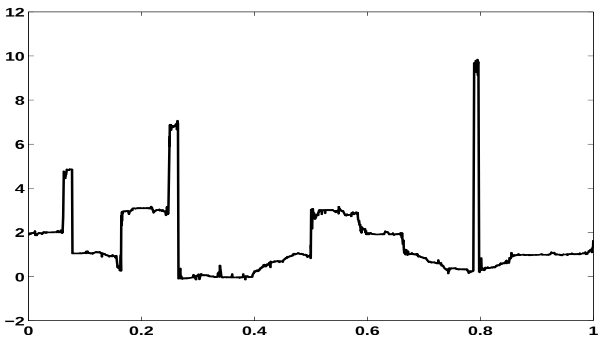

Figure 10: Output from an adaptive multiscale kernel transform, where two sided kernels are given priority above the one sided alternatives. This routine shows less falsely discovered singularities.

vestigate the possibility of different reconstruction schemes, including non-linear ones. Indeed, as the decomposition is overcomplete, the reconstruction is not unique. As for the actual denoising, we are interested in tree-structured coefficient selection, as well as in a smoothing that treats even and odd details in a possibly different way. This is because even details serve mainly perfect reconstruction, while the odd details describe the sparsity of the signal representation. An extension towards multiscale local polynomial smoothing is another objective for further research.

REFERENCES

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

[2] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo. Wavelet transforms that map integers to integers. *Appl. Comp. Harmon. Anal.*, 5(3):332–369, 1998.

[3] R. Claypoole, G. M. Davis, W. Sweldens, and R. Baraniuk. Nonlinear wavelet transforms for image coding via lifting. *IEEE Transactions on Image Processing*, 12(12):1449–1459, 2003.

[4] R. L. Claypoole, R.G. Baraniuk, and R. D. Nowak. Adaptive wavelet transforms via lifting. In *Proceedings of the 1998 International Conference on Acoustics, Speech, and Signal Processing - ICASSP '98*, 1998.

[5] I. Daubechies, I. Guskov, P. Schröder, and W. Sweldens. Wavelets on irregular point sets. *Phil. Trans. R. Soc. Lond. A*, 357:2397–2413, 1999.

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

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

[8] V. Delouille, M. Jansen, and R. von Sachs. Second generation wavelet methods for denoising of irregularly spaced data in two dimensions. *Signal Processing*, 86(7):1435–1450, 2006.

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

[10] G. Deslauriers and S. Dubuc. Symmetric iterative interpolation processes. *Constructive Approximation*, 5:49–68, 1989.

[11] I. Gijbels, A. Lambert, and P. Qiu. Edge-preserving image denoising and estimation of discontinuous surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1075–1087, 2006.

[12] M. Jansen. Stable edge-adaptive multiscale decompositions using updated normal offsets. *IEEE Transactions on Signal Processing*, 56(7):2718–2727, 2008.

[13] M. H. Jansen, G. P. Nason, and B. W. Silverman. Second generation wavelets and empirical bayes estimates for scattered data smoothing. Technical report, Bristol University, 2001.

[14] M. Knight and G.P. Nason. Improving prediction of hydrophobic segments along a transmembrane protein sequence using adaptive multiscale lifting. *SIAM Journal on Multiscale Modeling and Simulation*, 5:115–129, 2006.

[15] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[16] M. Nunes, M. Knight, and G.P. Nason. Adaptive lifting for nonparametric regression. *Statistics and Computing*, 16(2):143–159, 2006.

[17] P. J. Oonincx and P. M. de Zeeuw. Adaptive lifting for shape-based image retrieval. *Pattern Recognition*, 36(11):2663–2672, November 2003.

[18] G. Piella and H. J. A. M. Heijmans. Adaptive lifting schemes with perfect reconstruction. *IEEE Transactions on Signal Processing*, 50(7):1620–1630, July 2002.

[19] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, Box 812060, Wellesley MA 02181, fax 617-253-4358, 1996.

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

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

[22] W. Sweldens and P. Schröder. Building your own wavelets at home. In *Wavelets in Computer Graphics*, ACM SIGGRAPH Course Notes, pages 15–87. ACM, 1996.

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

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