CWI

Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

Adaptive Lifting Schemes with Perfect Reconstruction

G. Piella, H.J.A.M. Heijmans

Probability, Networks and Algorithms (PNA)

PNA-R0104 February 28, 2001

Report PNA-R0104
ISSN 1386-3711

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

Adaptive Lifting Schemes with Perfect Reconstruction

Gemma Piella

Henk J.A.M. Heijmans
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

ABSTRACT

In this paper, we propose a framework for constructing adaptive wavelet decompositions using the lifting scheme. A major requirement is that perfect reconstruction is possible without any overhead cost. In this paper we restrict ourselves to the update lifting stage. It is assumed that the update filter utilises local gradient information to adapt itself to the signal in the sense that smaller gradients 'evoke' stronger update filters. As a result, sharp transitions in a signal will not be smoothed to the same extent as regions which are more homogeneous. The approach taken in this paper differs from other adaptive schemes found in the literature in the sense that that no bookkeeping is required in order to have perfect reconstruction.

2000 Mathematics Subject Classification: 42C40, 94A12
Keywords and Phrases: Signal processing, perfect reconstruction filter bank, wavelet, lifting scheme, update lifting, adaptive filter, gradient-based decision, threshold criterion.
Note: This work was carried out under project PNA4.2 "Wavelets and Morphology". The research of the first author is sponsored by the Dutch Technology Foundation STW.

1. Introduction

Today, multiresolution decompositions are widely used in signal and image processing applications such as speech processing, image denoising, and image and video compression. The success of multiresolution techniques is largely due to the fact that signals often contain physically relevant features at many different scales or resolutions. Analysing such signals with transforms that try to unravel their multiscale structure may lead to a better understanding and give rise to more powerful processing techniques. Another major reason for taking recourse to multiresolution approaches is that the resulting algorithms may be more efficient.

A family of multiresolution decompositions which are invertible and non-redundant, i.e., the size of the input equals the size of the output, are discrete wavelet transforms, or perfect reconstruction filter banks as they are mostly called in the engineering literature. As such, the wavelet transform differs from various other multiresolution decompositions such as pyramids and frame decompositions, which are redundant, and scale-spaces and PDE-approaches, which are non-invertible in general. Both aforementioned properties of the wavelet transform, i.e., invertibility and non-redundancy, turn it into a highly efficient and applicable representation for a broad range of signal and image processing tasks such as denoising and, particularly, compression.

Originally, wavelet transforms were linear, and their construction was based on classical tools from functional and harmonic analysis such as the Fourier transform. The introduction of the lifting scheme by Sweldens [17–19] (but see also [1] for a related scheme, known as a "ladder network") opened the way to the design of nonlinear wavelet transforms [6,9,10,12,13]. Heijmans and Goutsias [14], for example, used the lifting scheme to build a class of nonlinear perfect reconstruction filter banks, which they called morphological wavelets.

In all these approaches, the flexibility and freedom offered by the lifting scheme was merely used to replace linear filters by nonlinear ones, such as those deriving from mathematical morphology. A severe limitation is that the filter structure is fixed, and thus cannot cope with the sudden changes in the input signal. In many applications, however, it is desirable to have a filter bank that somehow determines how to shape itself according to the data that it analyses. This can be achieved by allowing the lifting scheme to adapt its update and prediction filters to the local properties of the signal, thus giving rise to so-called *adaptive wavelets.*

The idea of adapting the wavelets to the local characteristics of the data is discussed in greater detail in Section 3, after we have given a brief reminder on the original lifting scheme in Section 2. In Section 4, we present a general framework for adaptive lifting schemes. In this paper, we concentrate ourselves on the update lifting step, and we assume that the update parameters depend point-wise on local gradient information. The update filters, as well as the necessary conditions for perfect reconstruction, are discussed in more detail in Section 5 and Section 6. In Section 7, we study the general scheme where we may have a continuous family of update operators depending on the l^1-norm of the gradient. In Section 8, we consider the binary case where the system switches between two update filters, depending on a threshold criterion for the gradient. Section 9 presents some experiment results to illustrate the performance and the potential of the adaptive schemes proposed here. Finally, Section 10 contains some concluding remarks and plans for future work.

2. The Lifting scheme

In this section we present a short reminder on the original lifting scheme as it was introduced by Sweldens [17–19]. An illustration of this scheme can be found in Fig. 1.

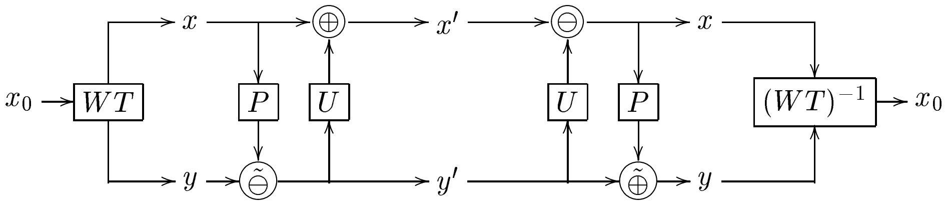

Figure 1: *General lifting scheme. The original signal x_0 is split into the approximation signal x and the detail signal y by a given wavelet transform WT. The prediction map P acting on x is used to modify y, resulting in a new detail signal y'. Subsequently, the update map U acting on y' is used to modify x, yielding a new approximation signal x'. At synthesis, the original signal x_0 is reconstructed by reversing the lifting steps and applying the inverse of WT.*

The lifting scheme is a very general and highly flexible tool for building new wavelet decompositions from existing ones. It can be applied with various goals in mind. A first objective might be to 'improve' a given wavelet decomposition, e.g., by increasing the number of vanishing moments it possesses. But from a completely different point of view, the lifting scheme can be exploited as a tool for building wavelets on irregular grids, e.g., a triangulation of some odd-shaped geometrical surface. Furthermore, it offers the possibility to replace linear filters by nonlinear ones [6, 9, 10, 12, 13], such as rank-order filters or morphological operators [14]. This paper is concerned with yet another application of the lifting scheme, viz. the design of adaptive wavelet decompositions.

A general lifting scheme comprises three main steps:

1. **Splitting:** The original signal x_0 is splitted into two parts, the *approximation signal* $x \in \mathcal{X}$, and the *detail signal* $y \in \mathcal{Y}$. In general, this partition is the outcome of a

particular wavelet transform. The most simple case is the one where the domain of the signal is subdivided into two disjoint subsets. In the one-dimensional case, this subdivision often comprises the even and odd samples, e.g., $x(n) = x_0(2n)$, $y(n) = x_0(2n+1)$. This latter decomposition is sometimes referred to as the *lazy wavelet transform.*

2. **Prediction lifting:** The detail signal y is predicted using information contained in the approximation signal x and is replaced by the prediction error

$$y' = y \mathbin{\tilde{\ominus}} P(x)\,,$$

where $P : \mathcal{X} \to \operatorname{Ran}(P)$ represents the prediction operator, and $\tilde{\ominus}$ is a binary difference operator mapping $\mathcal{Y} \times \operatorname{Ran}(P)$ into $\mathcal{Y}'$. The prediction error $y' \in \mathcal{Y}'$ becomes the new detail signal. To guarantee invertibility of the lifted wavelet transform, the difference operator needs to be invertible, i.e., there exists an addition operator $\tilde{\oplus}$ such that

$$y = (y \mathbin{\tilde{\ominus}} p) \mathbin{\tilde{\oplus}} p\,,$$

for all $y \in \mathcal{Y}$ and $p \in \operatorname{Ran}(P)$. We refer to this condition as the *reversibility condition.*

3. **Update lifting:** The approximation signal x is updated using information contained in the detail signal y':

$$x' = x \oplus U(y')\,.$$

Here $U : \mathcal{Y} \to \operatorname{Ran}(U)$ represents the update operator and $\oplus$ is a binary addition operator mapping $\mathcal{X} \times \operatorname{Ran}(U)$ into $\mathcal{X}'$. In this case the *reversibility condition* is said to hold if there exist a binary subtraction operator $\ominus$ so that

$$x = (x \oplus u) \ominus u\,, \tag{2.1}$$

for all $x \in \mathcal{X}$ and $u \in \operatorname{Ran}(U)$.

A general lifting scheme may comprise any sequence of update and prediction lifting steps. In practice, these lifting steps are chosen in such a way that the resulting decomposition is an 'improvement' of the original one. Here, the word 'improvement' can have various meanings. For example, the lifted wavelet may have more vanishing moments than the original one, or it may be better able to decorrelate the signals within a given class, etc.

In case that the reversibility conditions hold for all the addition and subtraction operators in the chain of lifting steps, then the resulting wavelet transform is invertible. One simply needs to reverse the order of the operations, use $\ominus$, $\tilde{\oplus}$ instead of $\oplus$, $\tilde{\ominus}$, and replace the initial splitting by a merging step.

Daubechies and Sweldens [5] have shown that all classical wavelet decompositions can be implemented using the lifting scheme. Furthermore, as we observed before, lifting does by no means require that the prediction and update operators are linear and/or fixed, nor that the underlying sampling grid is regular. With the lifting scheme, it becomes possible to build 'any wavelet you like' on 'every geometrical structure you are interested in'. In this paper, we will exploit this fact by adapting the update operator to the local properties of the signal.

3. Adaptive wavelets

The previous section shows clearly that there is an unlimited choice of wavelet transforms. An important reason for the success of wavelets is their intrinsic ability to approximate smooth signals efficiently; thus, by selecting the appropriate wavelet basis, one can approximate such signals with very high accuracy using only a limited number of coefficients. It is precisely this capacity of wavelets that makes them so useful for compression. Unfortunately, most signals that one encounters in the real world, including audio signals as well as imagery and video, are not as smooth as these approaches request. Classical wavelet transforms cannot get along very well with discontinuities encountered in real-world signals. Such singularities tend to give rise to large coefficients in their proximity, which is very undesirable in compression. To overcome such problems, one may take recourse to schemes which, in decomposing the signal, take into account its local properties, such as the local variance or gradient. In this case one often speaks about *adaptive* approaches. There are various ways to build adaptivity into the decomposition.

As observed, in many applications, the most interesting parts of a signal are its singularities. A useful idea, therefore, is to look for wavelets that are capable of 'tracking' the shape of the discontinuities. This had led to construction of functions whose support has a shape that can be adapted to the regularity of the signal being analysed. Donoho [7] studies the optimal approximations of particular classes of indicator functions with an overcomplete collection of atoms called *wedgelets*. His construction is based on a multiscale organisation of the edge data. Another construction due to Donoho [8] are the *ridgelets*. These are elongated wavelets especially suited for object discontinuities across straight lines. Recently, *bandelets* have been proposed by Le Pennec and Mallat [15]. They constructed orthonormal bases of wavelets which take advantage of the 'regularity of edges' in images. Singularities are first detected with so-called *foveal wavelets*, and then chained together to form edge curves. The foveal coefficients are then decomposed with standard wavelets bases. The resulting wavelets have their support in a band surrounding the edge curve, hence the name bandelet.

A second possibility is to choose a basis depending on the signal. The *best basis algorithm* [16] for example, selects a wavelet basis by minimising a concave cost function such as the entropy or an l^p-norm. In such an approach, the filter coefficients are fixed for an entire block of data as the optimisation criterion is a global one. Here we are interested in decompositions where the filter coefficients may vary locally, taking into account local variations of the signal. An interesting approach has been taken by Chan and Zhou [2]. Instead of changing the filter coefficients, they choose to change the input signal in the proximity of discontinuities through an extrapolation procedure. By recording these changes, the original signal can be recovered at synthesis.

Trappe and Liu [20] build adaptivity into the prediction step of the lifting scheme. Their aim is to design a data-dependent prediction filter to minimise the predicted detail signal. They distinguish two different approaches. Their first approach is global in the sense that the l^2-norm of the entire detail signal is minimised using Wiener filter theory. Their second approach uses a local optimisation criterion and, in this case, the coefficients of the prediction filter vary over time. Here the filter coefficients at a give time n are updated using the approximation signal x and the predicted detail y' at time $n-1$. In this scheme, perfect reconstruction is automatic. A very similar approach is used by Gerek and Çetin [11].

Claypoole *et al.* [3,4] propose an adaptive lifting scheme, which they call the *space-adaptive transform*, which lowers the order of the approximation near jumps to avoid predicting across discontinuities. This approach does not fit within the classical lifting scheme as the prediction step does require input from both channels. To guarantee perfect reconstruction at synthesis, one has to keep track of the filter choices made at each point. As a consequence, the resulting decomposition is no longer non-redundant in general.

Our approach resembles the approach by Claypoole *et al.* in the sense that it does not fit in the classical lifting scheme either. However, we try to develop our scheme in such a way that no bookkeeping is required. At the synthesis step we will still be able to recover the decision, i.e., the choice of the filter, made at the analysis step.

4. A general framework for adaptive lifting systems

In this section, we describe the adaptive lifting scheme in its upmost generality. Although in our terminology, we will speak about 'update lifting', the same framework can be applied to prediction lifting.

The basic idea underlying adaptive lifting is that the choice of the update operator U (but the same is true for prediction) depends on the information locally available within both the approximation signal x and the detail signal y. Recall that in the original lifting scheme, as discussed in Section 2, the update operator $U : \mathcal{Y} \to \mathrm{Ran}(U)$ is fixed, and that the output approximation signal equals

$$x' = x \oplus U(y)\,,$$

where $\oplus$ is a given addition operator mapping $\mathcal{X} \times \mathrm{Ran}(U)$ into $\mathcal{X}'$. In the adaptive case we assume that the choice of both the update operator U and the addition may be location-dependent. In fact, this choice is triggered by a so-called *decision map* $D : \mathcal{X} \times \mathcal{Y} \to \mathcal{D}^{\mathbb{Z}}$, where $\mathcal{D}$ is the *decision set.* In Fig. 2 we give a schematic representation of our lifting scheme. Thus, for every possible decision $d \in \mathcal{D}$ we have

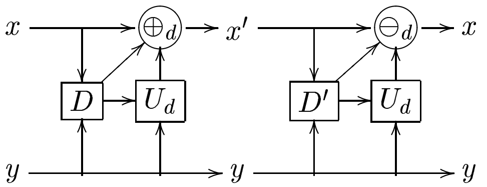

Figure 2: *Adaptive update lifting scheme.*

1. an update operator U_d which maps $\mathcal{Y}$ into a set $\mathrm{Ran}(U_d)$, the range of U_d;
2. a binary addition operator $\oplus_d$ which maps $\mathcal{X} \times \mathrm{Ran}(U_d)$ into $\mathcal{X}'$.

In this framework, the analysis step of the adaptive update lifting scheme looks as follows:

$$x'(n) = x(n) \oplus_{d_n} U_{d_n}(y)(n)\,, \qquad \forall n \in \mathbb{Z}\,, \tag{4.1}$$

where $d_n = D(x, y)(n)$ is the decision at location n.

It is evident that we also need to assume that the reversibility assumption in Section 2 holds for every possible decision $d \in \mathcal{D}$. Thus, for every $d \in \mathcal{D}$ there exists a binary operation $\ominus_d : \mathcal{X}' \times \mathrm{Ran}(U_d) \to \mathcal{X}$ such that

$$\big(x \oplus_d U_d(y)\big) \ominus_d U_d(y) = x\,,$$

for $(x, y) \in \mathcal{X} \times \mathcal{Y}$. In the non-adaptive case, where U and $\oplus$ are fixed, the analysis step can be inverted by (2.1), that is,

$$x(n) = x'(n) \ominus U(y)(n)\,.$$

In the adaptive case, however, we need to know d_n at every n to get perfect reconstruction. But $d_n = D(x, y)(n)$, and to compute it, it seems that we need to know the original signal x. In most cases, this is impossible, and hence perfect reconstruction is out of reach. But, as we will show later, there exist a number of situations in which it is still possible to recover d_n from a posteriori decision map D' which uses x' and y as input. Obviously, D' needs to satisfy

$$D'(x', y) = D(x, y)\,, \tag{4.2}$$

for $(x,y) \in \mathcal{X} \times \mathcal{Y}$, and with x' given by (4.1). We refer to this property as the *decision conservation condition.* It is obvious that this condition is satisfied if the decision map depends only on y, since then we can choose $D' = D$. For that reason, we reckon the case where D depends only on y among the non-adaptive lifting schemes. In the sequel we shall only consider schemes which are truly adaptive.

We start with a simple example.

4.1. Example. Let $\mathcal{X} = \mathcal{Y} = \mathrm{I\!R}^{\mathbb{Z}}$. Consider the three-valued decision map

$$D(x,y)(n) = \begin{cases} -1 & \text{if } |x(n) - y(n-1)| < |x(n) - y(n)| \\ +1 & \text{if } |x(n) - y(n-1)| > |x(n) - y(n)| \\ 0 & \text{if } |x(n) - y(n-1)| = |x(n) - y(n)| \,. \end{cases}$$

Thus, in this case, $D = \{-1, 0, +1\}$. Let $U_d,\ d \in \mathcal{D}$, be given by

$$U_d(y)(n) = \begin{cases} y(n-1) & \text{if } d = -1 \\ \frac{1}{2}(y(n-1) + y(n)) & \text{if } d = 0 \\ y(n) & \text{if } d = +1 \,. \end{cases}$$

Let $\oplus$ be independent of n, namely

$$(x \oplus y)(n) = \frac{1}{2}(x(n) + y(n)), \quad x, y \in \mathrm{I\!R}^{\mathbb{Z}} \,.$$

This scheme has the following simple interpretation. The updated sample $x'(n)$ is obtained by averaging the original sample $x(n)$ with its neighbour which is closest in value. If both neighbours $y(n-1)$, $y(n)$ are equally close, we average $x(n)$ with the average of both neighbours. If we define $D' = D$, then (4.2) holds as we will demonstrate below.

If $d_n = -1$, then

$$|x'(n) - y(n-1)| = \frac{1}{2}|x(n) - y(n-1)|$$

and

$$\begin{aligned} |x'(n) - y(n)| &= |\tfrac{1}{2}(x(n) + y(n-1)) - y(n)| \\ &= |x(n) - y(n) + \tfrac{1}{2}(y(n-1) - x(n))| \\ &\geq |x(n) - y(n)| - \tfrac{1}{2}|y(n-1) - x(n)| \\ &> \tfrac{1}{2}|x(n) - y(n-1)| = |x'(n) - y(n-1)| \,, \end{aligned}$$

which results in $D'(x', y) = -1$. If $d_n = +1$, a similar reasoning yields $|x'(n) - y(n)| < |x'(n) - y(n-1)|$, and thus $D'(x', y) = +1$. If $d_n = 0$, there are two possibilities:

(*i*) $y(n) = y(n-1)$: it is obvious that $|x'(n) - y(n-1)| = |x'(n) - y(n)|$, hence $D'(x', y) = 0$.

(*ii*) $x(n) - y(n) = y(n-1) - x(n)$: then $x'(n) = x(n)$, and therefore $D'(x', y)(n) = 0$.

Thus (4.2) holds and we get perfect reconstruction. To illustrate the performance of our adaptive system, we compare it to the non-adaptive scheme where D equals 0 for all n, that is, $U(y)(n) = \frac{1}{2}(y(n-1) + y(n))$ for all n. In both cases, the update step is followed by a non-adaptive prediction step. The prediction filter used is $P(z) = \frac{1}{2} + \frac{1}{2}z$, or equivalently, $y'(n) = y(n) - \frac{1}{2}(x(n) + x(n+1))$. The original as well as the decomposition signals are depicted in Fig. 3. The original test signal is shown on the top left, and the approximation and detail signals below. The middle row shows the decomposition results from the non-adaptive scheme, while the last row depicts the results from the adaptive scheme.

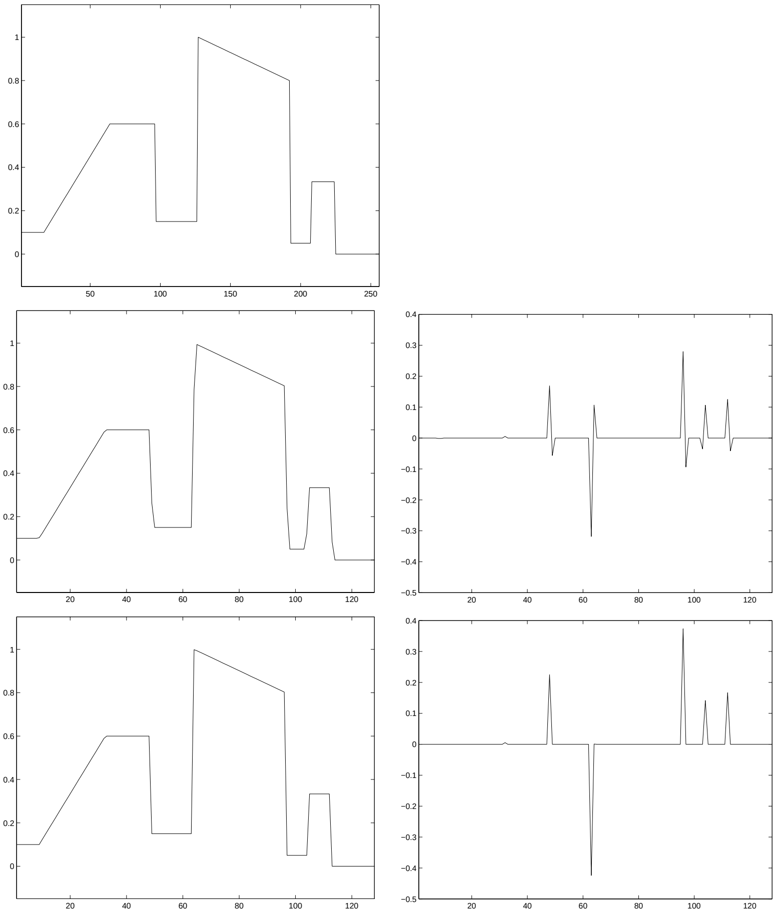

Figure 3: *The top image shows the original test signal. The second and third rows depict the decomposition signals when applying respectively the non-adaptive and the adaptive schemes discussed in Example 4.1. The images at the left represent the approximation signal, while the images at the right represent the detail signal.*

In the previous example, the decision $D(x, y)(n)$ at n depends on the sign of the expression $|x(n) - y(n-1)| - |x(n) - y(n)|$, which can be -1, 0, $+1$. Thus the decision depends on the value of the function S given by $S(x, y)(n) = |x(n) - y(n-1)| - |x(n) - y(n)|$, namely $D(x, y)(n) = \operatorname{sign} S(x, y)(n)$.

In the remainder of this paper we restrict ourselves to the case where the decision $d_n = D(x, y)(n)$ depends on the gradient at the point n, and more particularly on the l^1-norm. In Section 7 we investigate the case where d_n equals the l^1-norm of the gradient, and hence may assume any nonnegative value. In this case we have a continuous family of update operators. In Section 8, d_n can only assume the values 0 and 1, governed by a simple threshold criterion. In this case, we choose between two different update operators U_0 and U_1.

5. The update filters

Throughout the remainder of this paper we assume that the update operator U_d is a 2-tap filter of the form

$$U_d(y)(n) = \lambda_d y(n-1) + \mu_d y(n)\,, \tag{5.1}$$

where λ_d, μ_d are the filter coefficients. We also assume that the addition $\oplus_d$ is of the simple form

$$(x \oplus_d u)(n) = \alpha_d\bigl(x(n) + u(n)\bigr) .$$

At this point, we only require that $\alpha_d \neq 0$ in order to guarantee that the reversibility assumption stated in Section 2 holds. It is obvious that

$$(x \ominus_d u)(n) = \frac{1}{\alpha_d} x(n) - u(n) .$$

Now the analysis step in (4.1) gives that

$$x'(n) = \alpha_{d_n}\bigl(x(n) + \lambda_{d_n} y(n-1) + \mu_{d_n} y(n)\bigr) ,$$

where d_n is the decision at location n. Writing

$$\beta_d = \alpha_d \lambda_d \text{ and } \gamma_d = \alpha_d \mu_d ,$$

we can rewrite the previous expression as

$$x'(n) = \alpha_{d_n} x(n) + \beta_{d_n} y(n-1) + \gamma_{d_n} y(n) . \tag{5.2}$$

The synthesis step (presumed that d_n is known) is given by

$$x(n) = \frac{1}{\alpha_{d_n}}\bigl(x'(n) - \beta_{d_n} y(n-1) - \gamma_{d_n} y(n)\bigr) . \tag{5.3}$$

Define, for future reference,

$$\kappa_d = \alpha_d + \beta_d + \gamma_d .$$

From now on, we will restrict ourselves to decision maps which are based on the gradient vector

$$(v(n), w(n)) = (x(n) - y(n-1), y(n) - x(n)) .$$

More specifically, we assume that

$$D(x,y)(n) = d(v(n), w(n)) , \tag{5.4}$$

where $d : \mathbb{R} \times \mathbb{R} \to \mathcal{D}$. Observe that

$$v(n) + w(n) = y(n) - y(n-1)$$

does not depend on x. Therefore, if $d(v,w)$ depends only on $v+w$, we are in the non-adaptive situation. We can prove the following auxiliary result.

5.1. Lemma. *Consider a gradient-based decision map. In order to have perfect reconstruction it is necessary that κ_d is constant on every subset $\mathcal{D}(c) \subseteq \mathcal{D}$ given by $\mathcal{D}(c) = \{d(v,w) \mid v + w = c\}$, where $c \in \mathbb{R}$ is a constant.*

Proof. Assume that, for some $c \in \mathbb{R}$, we have $d_1, d_2 \in \mathcal{D}(c)$ such that $\kappa_{d_1} \neq \kappa_{d_2}$. Assume that (v_j, w_j) is such that $d(v_j, w_j) = d_j$ for $j = 1, 2$. Let the signals x_j, y_j be such that

$$y_j(n-1) = q\,,\ \ x_j(n) = q + v_j\ ,\ \ y_j(n) = q + v_j + w_j = q + c\,.$$

From (5.2) we get that

$$\begin{aligned} x_j'(n) &= \alpha_{d_j}(q + v_j) + \beta_{d_j} q + \gamma_{d_j}(q + c) \\ &= \kappa_{d_j}(q + v_j) - (\beta_{d_j} + \gamma_{d_j})(q + v_j) + \beta_{d_j} q + \gamma_{d_j}(q + c) \\ &= \kappa_{d_j} q + \kappa_{d_j} v_j - \beta_{d_j} v_j + \gamma_{d_j} w_j\,. \end{aligned}$$

If we choose q in such a way that

$$\kappa_{d_1}q + \kappa_{d_1}v_1 - \beta_{d_1}v_1 + \gamma_{d_1}w_1 = \kappa_{d_2}q + \kappa_{d_2}v_2 - \beta_{d_2}v_2 + \gamma_{d_2}w_2 ,$$

which is possible since $\kappa_{d_1} \neq \kappa_{d_2}$, then we get that $x_1'(n) = x_2'(n)$. Since $y_1(n-1) = y_2(n-1)$ and $y_1(n) = y_2(n)$, this implies that perfect reconstruction is not possible. □

Taking into account all the previous assumptions, it becomes possible to simplify the notation substantially by working with vectors rather than with signals. The key observation is that for updating the value $x(n)$ we only need to know its two neighbouring samples $y(n-1)$ and $y(n)$ along with $x(n)$ itself. Thus, we may represent the update operator as a mapping from $\mathbb{R}^2$ to $\mathbb{R}$. Instead of (5.1) we shall write

$$U_d(y,z) = \lambda_d y + \mu_d z .$$

Here (y,z) replaces the vector $(y(n-1), y(n))$ in (5.1). Instead of (5.2) we get

$$x' = \alpha_d x + \beta_d y + \gamma_d z , \tag{5.5}$$

and using the previous assumptions we find that the decision d in (5.5) is obtained from

$$d = d(v,w) = d(x-y, z-x) , \tag{5.6}$$

where

$$v = x - y \quad \text{and} \quad w = z - x .$$

The synthesis step in (5.3) can now be written as

$$x = \frac{1}{\alpha_d}\big(x' - \beta_d y - \gamma_d z\big) . \tag{5.7}$$

Putting

$$v' = x' - y \quad \text{and} \quad w' = z - x' ,$$

we derive from (5.5) that

$$\begin{aligned} v' &= (\alpha_d + \gamma_d)v + \gamma_d w + (\kappa_d - 1)y & (5.8)\\ w' &= \beta_d v + (\alpha_d + \beta_d)w - (\kappa_d - 1)z . & (5.9) \end{aligned}$$

Note that

$$v + w = v' + w' = z - y .$$

6. Perfect reconstruction condition

With the assumptions specified in the previous section we arrive at the following problem.

Problem statement. *Assume that we have a decision function $d : \mathbb{R} \times \mathbb{R} \to \mathcal{D}$, and filter coefficients $\alpha_d, \beta_d, \gamma_d \in \mathbb{R}$ for every $d \in \mathcal{D}$. Is the mapping*

$$(x,y,z) \mapsto (x', y, z) ,$$

where x' and d are given by (5.5) *and* (5.6) *respectively, invertible?*

We assume henceforth that the decision is given by the l^1-norm of the gradient:

$$d(v,w) = |v| + |w| . \tag{6.1}$$

In Lemma 5.1 we have seen that a necessary requirement for perfect reconstruction is that $\kappa_d = \alpha_d + \beta_d + \gamma_d$ is constant on every subset $\mathcal{D}(c)$. It is obvious that for the decision map in (6.1) we have $\mathcal{D}(0) = \mathbb{R}_+$. Thus we arrive at the following result.

6.1. Lemma. *Assume that the decision is given by* (6.1). *In order to have perfect reconstruction it is necessary that κ_d is constant for all $d \in \mathcal{D}$.*

Henceforth we assume that $\kappa_d = 1$, i.e.,

$$\alpha_d + \beta_d + \gamma_d = 1\,, \text{ for all } d \in \mathcal{D}\,.$$

In this case (5.8)-(5.9) can be rewritten as

$$\begin{aligned} v' &= (1-\beta_d)v + \gamma_d w & (6.2)\\ w' &= \beta_d v + (1-\gamma_d)w\,, & (6.3)\end{aligned}$$

which in matrix notation looks as follows:

$$\begin{pmatrix} v' \\ w' \end{pmatrix} = \begin{pmatrix} 1-\beta_d & \gamma_d \\ \beta_d & 1-\gamma_d \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix}\,. \tag{6.4}$$

The determinant of the matrix in (6.4) equals $1-\gamma_d-\beta_d = \alpha_d$. Assuming that

$$\alpha_d \neq 0 \ \text{ for } \ \text{all } d \in \mathcal{D}\,,$$

we can invert the equations in (6.2)-(6.3), and we get

$$\begin{aligned} v &= \frac{1}{\alpha_d}\Big((1-\gamma_d)v' - \gamma_d w'\Big)\\ w &= \frac{1}{\alpha_d}\Big(-\beta_d v' + (1-\beta_d)w'\Big)\,.\end{aligned}$$

However, in order to carry out this inversion we need to know the coefficients $\alpha_d, \beta_d, \gamma_d$ which all depend on $d = d(|v|+|w|)$. This gives rise to an equation for the unknown decision d. In order to have perfect reconstruction, this equation needs to have a unique solution for every gradient vector $(v,w) \in \mathbb{R}^2$.

In this paper we will consider two different cases. In the following section we consider the general case, i.e., the decision d equals the l^1-norm of the gradient, corresponding with a possibly infinite collection of update filters. In Section 8, we specialise to the case where there exists only two possible decisions $d=0$ and $d=1$, hence two update filters U_0 and U_1, depending on a threshold criterion for the gradient.

7. The general case

In this section we assume that the decision d equals the l^1-norm of the gradient. In the first proposition below we will give sufficient conditions for the filter coefficients $\alpha_d, \beta_d, \gamma_d$ which guarantee that we have perfect reconstruction.

7.1. Proposition. *Perfect reconstruction is possible in each of the following two cases:*

(a) $\alpha_d > 0$ for all $d \geq 0$, and β_d, γ_d are non-increasing with respect to d.

(b) $\alpha_d < 0$ for all $d \geq 0$, and β_d, γ_d are non-decreasing with respect to d.

Proof. We show that $x_1 \neq x_2$ implies that $x_1' \neq x_2'$ in both cases. Without lost of generality we may assume $z \geq y$ and that $x_2 > x_1$. A straightforward computation shows

$$\begin{aligned} x_2' - x_1' &= (\beta_1-\beta_2)\Delta + \alpha_1(w_1-w_2) + (\alpha_1-\alpha_2)w_2 & (7.1)\\ &= (\gamma_2-\gamma_1)\Delta + \alpha_2(v_2-v_1) + (\alpha_2-\alpha_1)v_1\,, & (7.2)\end{aligned}$$

where $\Delta = z - y$. We distinguish three different cases.

(i) $z > x_2 > x_1 > y$: In this case $d_1 = d_2 = \Delta$, which means that the filter coefficients are the same for both inputs. Thus, the first and last term of (7.1) are zero, and $x_2' - x_1' = \alpha_\Delta(w_1 - w_2) > 0$.

(ii) $x_2 > x_1 > z$ or $x_2 > z > x_1 > y$: Observe that in both cases $d_2 > d_1$, $w_2 < 0$, and $w_1 - w_2 > 0$. If $\alpha_d > 0$, and β_d, γ_d are non-increasing, then $(\beta_1 - \beta_2)\Delta > 0$, $\alpha_1(w_1 - w_2) > 0$, and $(\alpha_1 - \alpha_2)w_2 > 0$. Hence, we get from (7.1) that $x_2' - x_1' > 0$. If $\alpha_d < 0$, and β_d, γ_d are non-decreasing, then all terms in (7.1) are negative, and we get $x_2' - x_1' < 0$.

(iii) $x_2 > z > y > x_1$: In this situation we distinguish between the case where $d_2 \geq d_1$ and the case where $d_2 < d_1$. If $d_2 \geq d_1$, then $w_2 < 0$ and $w_1 - w_2 > 0$; we can use the same argument as in case *(ii)*. If $d_2 < d_1$, we use the identity in (7.2). Note that $v_1 < 0$ and $v_2 - v_1 > 0$. Consequently, if $\alpha_d > 0$, and β_d, γ_d are non-increasing, all terms in (7.2) are positive and we get $x_2' - x_1' > 0$. On the other hand, if $\alpha_d < 0$, and β_d, γ_d are non-decreasing, all terms in (7.2) are negative, and we get $x_2' - x_1' < 0$.

□

So far, we have only derived conditions which guarantee that perfect reconstruction is possible, but we have not yet given the correspondent algorithm. The lemma below will help us to construct such an algorithm. In this lemma we shall only deal with the first case in Proposition 7.1, i.e., we assume that $\alpha_d > 0$ for all $d \geq 0$, and β_d, γ_d are non-increasing with respect to d. Assume as before that we have the input and output triples (x, y, z) and (x', y, z), respectively, and that $z \geq y$. Put

$$\Delta = z - y \quad \text{and} \quad d = |x - y| + |z - x|\,,$$

that is, d is the l^1-norm of the gradient with respect to input triple (x, y, z). It is evident that

$$\Delta \leq d\,.$$

Furthermore, it is easy to show that

$$y + \gamma_d \Delta \leq y + \gamma_\Delta \Delta \leq z - \beta_\Delta \Delta \leq z - \beta_d \Delta\,.$$

We prove the following lemma.

7.2. Lemma. *Assume that $z \geq y$ and let $\Delta = z - y$ and $d = |x - y| + |z - x|$. The following relations hold:*

$$\begin{array}{ccc} x < y & \Longleftrightarrow & x' < y + \gamma_d \Delta \\ y \leq x \leq z & \Longleftrightarrow & y + \gamma_\Delta \Delta \leq x' \leq z - \beta_\Delta \Delta \\ x > z & \Longleftrightarrow & x' > z - \beta_d \Delta\,. \end{array}$$

Proof. Under the given assumptions we have $\Delta \leq d$. We can easily establish the following identities:

$$\begin{aligned} x' &= x + \gamma_d w - \beta_d v && (7.3)\\ &= z - \alpha_d w - \beta_d \Delta && (7.4)\\ &= y + \alpha_d v + \gamma_d \Delta\,. && (7.5) \end{aligned}$$

From we get immediately that iff $\alpha_d v < 0$, i.e., $v = x - y < 0$, then $x' < y + \gamma_d \Delta$. This proves the first relation. Similarly, (7.4) yields that $x' > z - \beta_d \Delta$ iff $\alpha_d w < 0$, that is, $w = z - x < 0$.

Now, if $y \leq x \leq z$, then $d = \Delta$, $v, w \geq 0$, and (7.4)-(7.5) yield that $y + \gamma_\Delta \Delta \leq x' \leq z - \beta_\Delta \Delta$. Conversely, assume that $y + \gamma_\Delta \Delta \leq x' \leq z - \beta_\Delta \Delta$, and suppose that $x < y$ (the case $x > z$ is treated analogously). We have seen above that this implies that $x' < y + \gamma_d \Delta$, which yields a contradiction. Hence $y \leq x \leq z$. This concludes the proof. □

Similar results can be obtained for case (b) of Proposition 7.1 as well as for the case that $z < y$.

The previous lemma is essential in the construction of an algorithm which performs the inversion step. Again, we restrict ourselves to the case $y \leq z$. Observe first that

$$x \in [y, z] \iff x' \in [y + \gamma_\Delta \Delta, z - \beta_\Delta \Delta]\,.$$

Thus, if $x' \in [y + \gamma_\Delta \Delta, z - \beta_\Delta \Delta]$, then

$$d = \Delta = z - y\,,$$

and reconstruction becomes straightforward. If, however, $x' \notin [y + \gamma_\Delta \Delta, z - \beta_\Delta \Delta]$, then

$$d = |y + z - 2x| = |y + z - \frac{2}{\alpha_d}(x' - \beta_d y - \gamma_d z)|\,,$$

where we have used (5.7). This can be rewritten as

$$d \cdot \alpha_d = |y + z - 2x' + (\beta_d - \gamma_d)(y - z)|\,.$$

This latter equation has a unique solution d, and having found this, reconstruction becomes straightforward. The other cases (Proposition 7.1(b) and/or $z < y$) can be treated similarly, and we arrive at the following algorithm.

7.3. Algorithm.

1. Compute $\Delta = |z - y|$.
2. Compute coefficients $\alpha_\Delta, \beta_\Delta, \gamma_\Delta$.
3. Compute the lower and upper limits, Y and Z, as

$$Y = \min(y + \gamma_\Delta(z - y), z - \beta_\Delta(z - y)) \qquad Z = \max(y + \gamma_\Delta(z - y), z - \beta_\Delta(z - y))\,.$$

4. If $x' \in [Y, Z]$ put

$$\gamma = \gamma_\Delta \quad \text{and} \quad \beta = \beta_\Delta$$

else

(4a) compute d by solving

$$d \cdot \alpha_d = |y + z - 2x' + (\beta_d - \gamma_d)(y - z)|$$

(4b) put

$$\gamma = \gamma_d \quad \text{and} \quad \beta = \beta_d\,.$$

5. Compute x from

$$x = \frac{x' - \beta y - \gamma z}{1 - \beta - \gamma}.$$

7.4. Example. Consider the case where

$$\gamma_d = \beta_d = \frac{\gamma_0}{\sigma d + 1}, \tag{7.6}$$

where $0 \leq \gamma_0 < \frac{1}{2}$ and $\sigma > 0$. It follows immediately that the conditions in Proposition 7.1(a) are satisfied. Steps (1) to (3) of Algorithm 1 are straightforward, and they yield the coefficients

$$\gamma_\Delta = \beta_\Delta = \frac{\gamma_0}{\sigma\Delta + 1} = \frac{\gamma_0}{\sigma|z-y| + 1}.$$

After computing the boundaries Y and Z, we have to check whether x' belongs to the interval $[Y, Z]$. If it does, we know that $d = \Delta$, and that $\gamma = \beta = \gamma_\Delta$. Now we can retrieve x following step (5). Otherwise, we must solve the equation given in step (4a) with $\alpha_d = 1 - 2\gamma_d$:

$$d \cdot (1 - 2\gamma_d) = |y + z - 2x'|.$$

Expressing γ_d as a function of d, and denoting $r = |y + z - 2x'|$, we arrive at the quadratic equation

$$\sigma d^2 + (1 - 2\gamma_0 - r\sigma)d - r = 0.$$

This equation has a unique positive solution:

$$d = \frac{-(1 - 2\gamma_0 - r\sigma) + \sqrt{(1 - 2\gamma_0 - r\sigma)^2 + 4r\sigma}}{2\sigma}.$$

From this d we can compute the filter coefficients $\gamma = \beta := \gamma_d$, and retrieve x using step (5).

8. Binary decisions based on a threshold criterion

In this section, we investigate a simple class of adaptive update lifting schemes where the decision map is based on a simple threshold criterion. To be precise, we assume that $D = \{0, 1\}$, and that the function d in (5.4) has the form

$$d(v, w) = \begin{cases} 1, & \text{if } |v| + |w| > T \\ 0, & \text{if } |v| + |w| \leq T, \end{cases} \tag{8.1}$$

where T is the threshold. Instead of (8.1) we sometimes use the shorthand notation

$$d(v, w) = [\,|v| + |w| > T\,], \tag{8.2}$$

where $[P]$ returns 1 if the predicate P is true, and 0 if it is false.

This binary scheme can be regarded as a particular instance of the case presented in the previous section, with coefficients given by

$$\beta_d = \begin{cases} \beta_0 & \text{if } d \leq T \\ \beta_1 & \text{if } d > T \end{cases} \qquad \gamma_d = \begin{cases} \gamma_0 & \text{if } d \leq T \\ \gamma_1 & \text{if } d > T \end{cases}$$

Assuming $\alpha_d + \beta_d + \gamma_d = 1$ for $d = \{0, 1\}$, Proposition 7.1 yields the following results.

8.1. Proposition. *Perfect reconstruction is guaranteed in each of the following two cases:*

(*a*) $\alpha_0 > 0$, $\beta_0 \geq \beta_1$, $\gamma_0 \geq \gamma_1$.

(*b*) $\alpha_1 < 0\,,\ \beta_0 \le \beta_1\,,\ \gamma_0 \le \gamma_1$.

In the previous section it was necessary to recover $d = |v| + |w|$ at the synthesis step. Here, it suffices to check if either d was smaller or equal than the threshold T, or if d was larger than T. In fact, we are interested in finding a, possibly different, threshold value T' which can be used to retrieve d at synthesis:

$$d = [\,|v'| + |w'| > T'\,]\,. \tag{8.3}$$

Recall from (6.4) that

$$\begin{pmatrix} v' \\ w' \end{pmatrix} = \begin{pmatrix} 1-\beta_d & \gamma_d \\ \beta_d & 1-\gamma_d \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix}\,, \tag{8.4}$$

which can be written in vector notation as $\boldsymbol{v'} = \boldsymbol{A}_d\boldsymbol{v}$. Let us henceforth denote by $\|\boldsymbol{v}\|$ the l^1-norm of $\boldsymbol{v}$, that is, $\|\boldsymbol{v}\| = |v| + |w|$ if $\boldsymbol{v} = (v, w)$. With this notation the threshold criterion in (8.3) can be reformulated as follows: given $T > 0$, there exists a $T' > 0$ such that

(*i*) if $\|\boldsymbol{v}\| \le T$, then $\|\boldsymbol{A}_0\boldsymbol{v}\| \le T'$,

(*ii*) if $\|\boldsymbol{v}\| > T$, then $\|\boldsymbol{A}_1\boldsymbol{v}\| > T'$.

We can prove the following result.

8.2. Lemma. *The threshold criterion holds if and only if*

$$\|\boldsymbol{A}_1^{-1}\|^{-1} \ge \|\boldsymbol{A}_0\|\,. \tag{8.5}$$

Proof. First we prove the 'if'-statement. We show that (*i*) and (*ii*) are satisfied if we put $T' = \|\boldsymbol{A}_0\|T$. First, if $\|\boldsymbol{v}\| \le T$ then $\|\boldsymbol{A}_0\boldsymbol{v}\| \le \|\boldsymbol{A}_0\| \cdot \|\boldsymbol{v}\| \le \|\boldsymbol{A}_0\|T = T'$, which proves (*i*). To prove (*ii*) we use that

$$\|\boldsymbol{v}\| = \|\boldsymbol{A}_1^{-1}\boldsymbol{A}_1\boldsymbol{v}\| \le \|\boldsymbol{A}_1^{-1}\| \cdot \|\boldsymbol{A}_1\boldsymbol{v}\|\,, \boldsymbol{v} \in \mathbb{R}^2\,.$$

From this we derive that

$$\|\boldsymbol{A}_1\boldsymbol{v}\| \ge \|\boldsymbol{A}_1^{-1}\|^{-1}\|\boldsymbol{v}\| \ge \|\boldsymbol{A}_0\| \cdot \|\boldsymbol{v}\|\,.$$

Thus, if $\|\boldsymbol{v}\| > T$, then $\|\boldsymbol{A}_1\boldsymbol{v}\| > \|\boldsymbol{A}_0\|T = T'$, and this proves the result.

To establish the 'only if' part, assume that (*i*)-(*ii*) hold. Choosing $T = 1$ we get from (*i*) that $\|\boldsymbol{A}_0\| \le T'$. Now (*ii*) reads as: "if $\|\boldsymbol{v}\| > 1$, then $\|\boldsymbol{A}_1\boldsymbol{v}\| > T'$". We estimate $\|\boldsymbol{A}_1^{-1}\|$. Choose $\boldsymbol{w} \in \mathbb{R}^2 \setminus \{0\}$ and put $\boldsymbol{v} = (1+\epsilon)\boldsymbol{A}_1^{-1}\boldsymbol{w}/\|\boldsymbol{A}_1^{-1}\boldsymbol{w}\|$, where $\epsilon > 0$. Obviously, $\|\boldsymbol{v}\| = 1 + \epsilon > 1$ and (*ii*) yields that $\|\boldsymbol{A}_1\boldsymbol{v}\| > T'$, that is, $(1+\epsilon)\|\boldsymbol{w}\|/\|\boldsymbol{A}_1^{-1}\boldsymbol{w}\| > T'$. This can be rewritten as

$$\|\boldsymbol{A}_1^{-1}\boldsymbol{w}\| < \frac{(1+\epsilon)}{T'}\|\boldsymbol{w}\| \le \frac{(1+\epsilon)}{\|\boldsymbol{A}_0\|} \cdot \|\boldsymbol{w}\|\,.$$

Thus, we conclude that $\|\boldsymbol{A}_1^{-1}\| \le (1+\epsilon)\|\boldsymbol{A}_0\|^{-1}$, for every $\epsilon > 0$. Therefore, $\|\boldsymbol{A}_1^{-1}\| \le \|\boldsymbol{A}_0\|^{-1}$ whence (8.5) follows immediately. □

The l^1-norm of the matrix $\boldsymbol{A}_d$ in (8.4) is given by

$$\|\boldsymbol{A}_d\| = \max\Big(|\beta_d| + |1-\beta_d|, |\gamma_d| + |1-\gamma_d|\Big)\,,$$

and the norm of its inverse equals

$$\|\boldsymbol{A}_d^{-1}\| = \frac{1}{|\alpha_d|}\max\Big(|\gamma_d| + |1-\beta_d|, |\beta_d| + |1-\gamma_d|\Big)\,.$$

Therefore, the condition $\|\boldsymbol{A}_1^{-1}\|^{-1} \geq \|\boldsymbol{A}_0\|$ can be written as

$$|\alpha_1|\Big(\max\big(|\gamma_1| + |1-\beta_1|, |\beta_1| + |1-\gamma_1|\big)\Big)^{-1} \geq \max\big(|\beta_0| + |1-\beta_0|, |\gamma_0| + |1-\gamma_0|\big) . \quad (8.6)$$

Putting

$$m_1 = \max\big(|\gamma_1| + |1-\beta_1|, |\beta_1| + |1-\gamma_1|\big) \quad \text{and} \quad m_0 = \max\big(|\beta_0| + |1-\beta_0|, |\gamma_0| + |1-\gamma_0|\big) ,$$

we can rewrite (8.6) as

$$m_0 m_1 \leq |\alpha_1| .$$

It is easy to show that $m_0 \geq 1$ and that $m_1 \geq |\alpha_1|$. Therefore, the only way to satisfy (8.6) is to choose $m_0 = 1$ and $m_1 = |\alpha_1|$. The first equality yields

$$\beta_0, \gamma_0 \in [0, 1] ,$$

and the second equality gives

$$\text{either} \quad \beta_1, \gamma_1 \leq 0 \quad \text{or} \quad \beta_1, \gamma_1 \geq 1 .$$

To establish this latter result we have used that $\alpha_1 = 1 - \beta_1 - \gamma_1$. We summarise our results in the following proposition.

8.3. Proposition. *Assume that the decision map is given by the threshold criterion in* (8.2). *At synthesis, the decision can be recomputed from the threshold criterion in* (8.3) *with* $T' = T$ *iff the following holds:*

(i) $0 \leq \beta_0, \gamma_0 \leq 1$*;*

(ii) either $\beta_1, \gamma_1 \leq 0$ *or* $\beta_1, \gamma_1 \geq 1$.

9. Simulation results

In this section, we present some preliminary results for the adaptive schemes presented in Section 7 and Section 8. In our future work we will present more elaborate experiments and make comparisons with other approaches found in the literature; see Section 3. To illustrate the performance of our adaptive schemes, we compare these schemes with their non-adaptive counterparts. In all cases, the update step is followed by a fixed prediction step of the form

$$y'(n) = y(n) - \Big(\frac{x'(n) + x'(n+1)}{2}\Big) .$$

The test signal (see top left of Fig. 4) is obtained by adding normally distributed noise to the undisturbed signal in Fig. 3.

First, we consider the threshold-based adaptive scheme described in Section 8. We choose $\beta_0 = \gamma_0 = \frac{1}{3}$, and $\beta_1 = \gamma_1 = 0$. Thus, the resulting low-pass filters are the average filter for $d = 0$, and the trivial filter (or the identity map) for $d = 1$. The approximation and detail signals, x' and y', are depicted in the second row of Fig. 4; here we have taken a threshold $T = 0.2$. The corresponding decision map is shown at the top right. This map shows that we make decision 1 at five positions (note that at these positions the gradient exceeds our threshold T). Anywhere else, the decision equals 0, meaning that the average update filter is used there. The decomposition signals obtained for both non-adaptive cases corresponding with $d = 0$ and $d = 1$ are shown respectively in the third and fourth rows of Fig. 4.

In this particular example, the adaptive scheme performs as the fixed scheme with $d = 0$ most of the time except for those five locations where the gradient exceeds the threshold T;

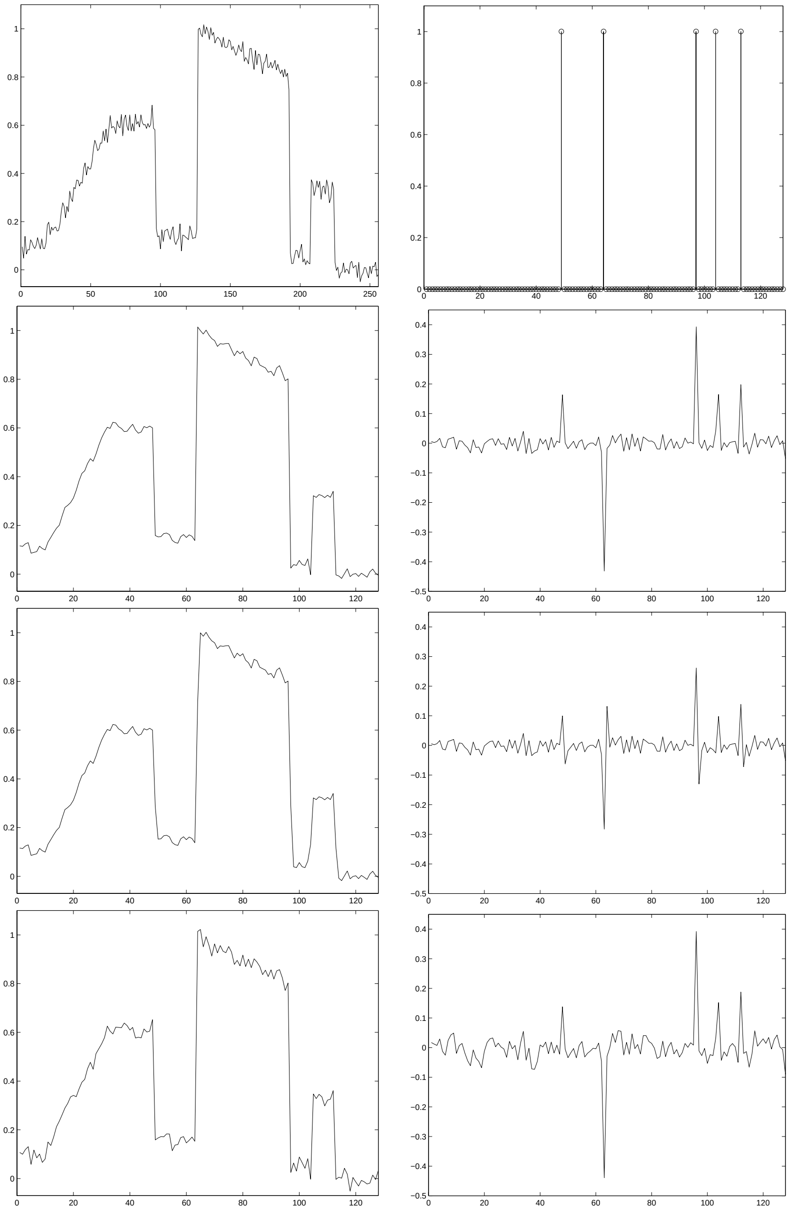

Figure 4: *The top left image shows the original test signal. The top right depicts the decision map for $T = 0.2$. The second row shows the corresponding approximation (left) and detail (right) signals. The third and fourth row correspond with the non-adaptive decompositions for $d = 0$ and $d = 1$, respectively.*

the scheme 'decides' that these locations correspond with sharp transitions in the signal and it does not apply any smoothing. Therefore, the adaptive scheme is capable of 'recognizing' the edges and preserving them, while simultaneously smoothing the more homogeneous regions. As a consequence, the detail signal remains small except near discontinuities. There, the detail signal takes the same value as in the non-adaptive case corresponding with $d = 1$ and, as a result, it avoids the double-peaked detail that one encounters in the non-adaptive case $d = 0$.

By varying the threshold T, the resulting system can be tuned to the one or the other behaviour. If T is very small, the adaptive system will behave more or less as the non-adaptive scheme with $d = 1$. If T is increased, the decomposition map will attain the value 0 more often, meaning that it behaves increasingly as the non-adaptive scheme corresponding to $d = 0$.

To get a first impression of the amount of compression that can possibly be achieved by the application of the adaptive scheme, we compute the entropies of the three detail signals in Fig. 4 and found an entropy of 4.51 for the adaptive scheme, 5.06 for the fixed scheme with $d = 0$, and 5.15 for the fixed scheme with $d = 1$. Note that a lower entropy may eventually lead to a higher compression ratio. However, we emphasise that these results are very preliminary and that a more thorough study needs to be carried out before we can reach any final conclusions.

Next, we present some results for the general scheme of Section 7. In particular, we consider the case where the coefficients are given by (7.6) in Example 7.4, with $\gamma_0 = \frac{1}{3}$ and $\sigma = 5$. That is,

$$\gamma_d = \beta_d = \frac{\frac{1}{3}}{5d+1} = \frac{1}{15d+3}\,.$$

Note that for small values of d, the resulting low-pass filter approximates the average filter ($\alpha = \beta = \gamma = \frac{1}{3}$), while for large values it behaves more like the trivial filter ($\alpha = 1, \beta = \gamma = 0$). Thus, again we can consider these two extreme cases to be the non-adaptive counterparts of the proposed scheme. We use the same input as in Fig. 4. The two bottom rows of Fig. 4 correspond with the two aforementioned extreme cases. The approximation and detail signals resulting from our scheme are depicted at the bottom row of Fig. 5. The corresponding decision map, which in this case equals the l^1-norm gradient, is shown at the top left. On the right, the corresponding coefficients $\gamma_d = \beta_d$ have been depicted. These figures show clearly that sharp transitions are smoothed to a much lesser degree than more homogeneous parts of the signal.

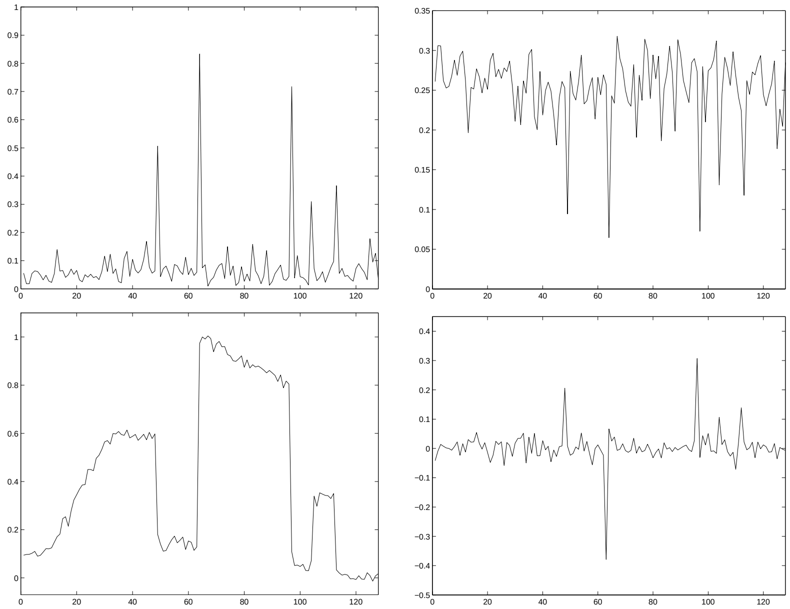

Figure 5: *First row: decision map d (left) and coefficients γ_d Right). Second row: approximation (left) and detail (right) signal.*

For this adaptive scheme, the entropy of the detail signal is 4.82, which is higher than the entropy we get in the threshold-based scheme (4.51), but lower than those obtained in the non-adaptive cases (5.06 and 5.15, respectively). A preliminary conclusion therefore is that wavelet decompositions using adaptive update lifting yield detail signals with lower entropies than in the non-adaptive case.

10. Conclusion

We have introduced a new framework for adaptive lifting. The basic idea underlying this scheme is to choose the lifting filters according to some decision criterion which may depend on local characteristics of the signal, such as (the norm of) the gradient. Our scheme distinguishes itself from other schemes like the one by Claypoole *et al.* [3, 4] in the sense that it does not require any bookkeeping. Although we have restricted ourselves to the update lifting step, the same framework can be applied for the prediction stage.

Preliminary experiments seem to indicate that these adaptive decompositions yield detail signals with lower entropies than in the non-adaptive case. This, together with the fact that bookkeeping is not needed, suggest that our adaptive schemes may be useful for compression. We intend to investigate this more thoroughly in the future.

In the literature there exist several wavelet decomposition schemes which, in various ways, try to take into account discontinuities (e.g., singularities in signals, edges in images) [2–4, 7, 8, 15]. We have briefly discussed some of these schemes in Section 3. We intend to investigate and understand the similarities and differences between these techniques and our own adaptive scheme. Other work in progress concerns the extension of our results to other decision criteria, to filters with more taps, and to the two-dimensional case. Finally, we want to examine cases where the linear update (and prediction) filters are replaced by morphological ones [14].

References

1. Bruekers, F. A. M. L., and van den Enden, A. W. M. New networks for perfect inversion and perfect reconstruction. *IEEE Journal on Selected Areas in Communications 10* (1992), 130–137.
2. Chan, T., and Zhou, H. Adaptive ENO-wavelet transforms for discontinuous functions. Tech. Rep. 21, Computational and Applied Mathematics, UCLA, Los Angeles, 1999.
3. Claypoole, R. L., Baraniuk, R. G., and Nowak, R. D. Adaptive wavelet transforms via lifting. Technical Report 9304, Department of Electrical and Computer Engineering, Rice University, Houston, Texas, April 1999.
4. Claypoole, R. L., Davis, G., Sweldens, W., and Baraniuk, R. D. Nonlinear wavelet transforms for image coding. In *Proceedings of the 31st Asilomar Conference on Signals, Systems, and Computers, Volume 1* (1997), pp. 662–667.
5. Daubechies, I., and Sweldens, W. Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications 4*, 3 (1998), 245–267.
6. de Queiroz, R. L., Florêncio, D. A. F., and Schafer, R. W. Nonexpansive pyramid for image coding using a nonlinear filterbank. *IEEE Transactions on Image Processing 7* (1998), 246–252.
7. Donoho, D. L. Wedgelets: nearly minimax estimation of edges. Tech. rep., Statistics Department, Stanford University, 1997.
8. Donoho, D. L. Orthonormal ridgelets and linear singularities. Tech. rep., Statistics Department, Stanford University, 1998.
9. Egger, O., Li, W., and Kunt, M. High compression image coding using an adaptive morphological subband decomposition. *Proceedings of the IEEE 83* (1995), 272–287.
10. Florêncio, D. A. F., and Schafer, R. W. A non-expansive pyramidal morphological image coder. In *Proceedings of the IEEE International Conference on Image Processing* (Austin, Texas, 1994), pp. 331–335.
11. Gerek, O. N., and Çetin, A. E. Adaptive polyphase subband decomposition structures for image compression. *IEEE Transactions on Image Processing 9* (October 2000), 1649–1659.
12. Hampson, F. J., and Pesquet, J.-C. A nonlinear subband decomposition with perfect reconstruction. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (Atlanta, Georgia, May 7-10, 1996), pp. 1523–1526.
13. Hampson, F. J., and Pesquet, J.-C. M-band nonlinear subband decompositions with perfect reconstruction. *IEEE Transactions on Image Processing 7* (1998), 1547–1560.
14. Heijmans, H. J. A. M., and Goutsias, J. Nonlinear multiresolution signal decomposition schemes: Part II: morphological wavelets. *IEEE Transactions on Image Processing 9*, 11 (2000), 1897–1913.
15. Le Pennec, E., and Mallat, S. Image compression with geometrical wavelets. In *IEEE Conference on Image Processing (ICIP)* (Vancouver, September 2000).

16. Mallat, S. *A Wavelet Tour of Signal Processing.* Academic Press, San Diego, California, 1998.

17. Sweldens, W. The lifting scheme: A new philosophy in biorthogonal wavelet constructions. In *Wavelet Applications in Signal and Image Processing III* (1995), A. F. Lain and M. Unser, Eds., Proceedings of SPIE, vol. 2569, pp. 68–79.

18. Sweldens, W. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis 3* (1996), 186–200.

19. Sweldens, W. The lifting scheme: A construction of second generation wavelets. *SIAM Journal of Mathematical Analysis 29* (1997), 511–546.

20. Trappe, W., and Liu, K. Adaptivity in the lifting scheme. In *33th Conference on Information Sciences and Systems* (Baltimore, March 1999), pp. 950–955.