Adaptive wavelets and their applications to image fusion and compression
Piëlla, G.

Citation for published version (APA):
Chapter 4

Adaptive update lifting: specific cases

In the previous chapter we have presented an axiomatic framework for adaptive wavelets constructed by means of an adaptive update lifting step. The adaptivity consists hereby that the update filter coefficients are triggered by a decision parameter. In particular, we have studied the case where this decision is binary and is obtained by thresholding the seminorm of a local gradient vector computed from the input signals to the system. The lifting scheme can therefore choose between two different update linear filters: if the seminorm of the gradient is above the threshold, it chooses one filter, otherwise it chooses the other. At synthesis, the decision is obtained in the same way but using the gradient computed from the bands available at synthesis (the updated approximation band and the unmodified detail bands). With such a thresholding-decision scheme, perfect reconstruction amounts to the threshold criterion, which says that the seminorm of the gradient at synthesis should be above the threshold only if the seminorm of the original gradient is. In Section 3.5, we stated necessary and sufficient conditions for the threshold criterion to hold.

In this chapter, we investigate perfect reconstruction conditions for several decision scenarios. First, we assume a binary decision map and linear filters as described in Section 3.4. We analyze different seminorms and derive sufficient conditions for perfect reconstruction stated in terms of the filter coefficients. We study the weighted seminorm in Section 4.1, the quadratic seminorm in Section 4.2, and the $l^1$-norm as well as the $l^\infty$-norm in Section 4.3. In Section 4.4, however, we consider the case where the decision map is not binary but continuous. In particular, we investigate the case where the decision equals the $l^1$-norm of the gradient vector, corresponding with a possibly infinite collection of update filters. In Section 4.5, we consider other alternatives which do not fit either into the specific cases previously described, but which extend the general framework proposed in Section 3.2 by allowing other decision maps and update filters.
4.1 Weighted gradient seminorm

4.1.1 Perfect reconstruction conditions

Recall that, for the choice of decision map and update filters we have made so far, the update lifting step can be written abstractly by

\[
\begin{align*}
\mathbf{v}' &= A_d \mathbf{v} \\
d &= \lfloor p(\mathbf{v}) > T \rfloor,
\end{align*}
\]

where \( d \in \{0, 1\} \) is the decision parameter which triggers the update step:

\[
\mathbf{x}' = \alpha_d \mathbf{x} + \sum_{j=1}^{N} \beta_{d,j} \mathbf{y}_j.
\]

Recall also that we assume

\[
\alpha_0 + \sum_{j=1}^{N} \beta_{0,j} = \alpha_1 + \sum_{j=1}^{N} \beta_{1,j} = 1
\]

with \( \alpha_d \neq 0 \) for both \( d = 0, 1 \), and \( \beta_{0,j} \neq \beta_{1,j} \) for some \( j \in \{1, \ldots, N\} \) (adaptivity condition for the update filters; see (3.14)). In this framework, perfect reconstruction is guaranteed if conditions (3.21)-(3.22) are satisfied. We remind that these are necessary and sufficient conditions for the threshold criterion to hold, hence sufficient conditions for perfect reconstruction.

Moreover, since \( A_d = I - \mathbf{u}\beta_d^T \) is invertible, we can rewrite these conditions as

\[
\begin{align*}
p(A_0) &< \infty \quad \text{and} \quad p(A_1^{-1}) < \infty \quad (4.2) \\
p(A_0)p(A_1^{-1}) &\leq 1. \quad (4.3)
\end{align*}
\]

In this section, we consider the situation where \( p \) is given by

\[
p(\mathbf{v}) = |\alpha^T \mathbf{v}|, \quad \text{with} \quad \alpha \neq 0.
\]

Since \( \mathbf{v} \) is a gradient vector, we call this seminorm the weighted gradient seminorm. Note that the adaptivity condition \( p(\mathbf{u}) > 0 \) holds if and only if \( \alpha^T \mathbf{u} \neq 0 \), i.e.,

\[
\sum_{j=1}^{N} a_j \neq 0.
\]

We establish necessary and sufficient conditions for the threshold criterion to hold.

**Lemma 4.1.1.** Let \( p \) be the weighted gradient seminorm defined in (4.4) and let \( A \) be the matrix \( A = I - \mathbf{u}\beta^T \), where \( \mathbf{u}, \beta \in \mathbb{R}^N \).

(a) \( \alpha^T \mathbf{u} = 0 \) implies \( p(A) = 1 \).
4.1. Weighted gradient seminorm

(b) Assume \( \mathbf{a}^T \mathbf{u} \neq 0 \),

(i) if \( \mathbf{a}, \mathbf{b} \) are collinear, then \( p(A) = |\alpha| \), where \( \alpha = 1 - \beta^T \mathbf{u} \);

(ii) if \( \mathbf{a}, \mathbf{b} \) are not collinear, then \( p(A) = \infty \).

Proof. From the definition of a matrix seminorm (see page 50) we have

\[
p(A) = \sup\{|\mathbf{a}^T \mathbf{A} \mathbf{v}| \mid \mathbf{v} \in \mathbb{R}^N \text{ and } |\mathbf{a}^T \mathbf{v}| = 1\}.
\]

Therefore, in order to calculate this seminorm we have to find the supremum of \( |\mathbf{a}^T \mathbf{A} \mathbf{v}| \) under the constraint \( |\mathbf{a}^T \mathbf{v}| = 1 \).

Assume \( \mathbf{a}^T \mathbf{u} = 0 \). Then,

\[
|\mathbf{a}^T \mathbf{A} \mathbf{v}| = |\mathbf{a}^T (I - \mathbf{u} \mathbf{b}^T) \mathbf{v}| = |\mathbf{a}^T \mathbf{v} - \mathbf{a}^T \mathbf{u} \mathbf{b}^T \mathbf{v}| = 1.
\]

This proves (a).

Now, assume \( \mathbf{a}^T \mathbf{u} \neq 0 \). We distinguish two cases, namely \( \mathbf{b} \) and \( \mathbf{a} \) are or are not collinear.

(i) \( \mathbf{b} \) collinear with \( \mathbf{a} \). In this case we can write \( \mathbf{b} = \gamma \mathbf{a} \) for some constant \( \gamma \in \mathbb{R} \) and we get

\[
|\mathbf{a}^T \mathbf{A} \mathbf{v}| = |\mathbf{a}^T (I - \mathbf{u} \mathbf{b}^T) \mathbf{v}| = |\mathbf{a}^T \mathbf{v} - \mathbf{a}^T \mathbf{u} \gamma \mathbf{a}^T \mathbf{v}| = |1 - \gamma \mathbf{a}^T \mathbf{u}| |\mathbf{a}^T \mathbf{v}| = |1 - \beta^T \mathbf{u}| = |\alpha|.
\]

This yields that \( p(A) = |\alpha| \).

(ii) \( \mathbf{b} \) not collinear with \( \mathbf{a} \). In this case we can express \( \mathbf{b} = \gamma \mathbf{a} + \mathbf{c} \) with \( \mathbf{a}^T \mathbf{c} = 0 \) and \( \mathbf{c} \neq 0 \).

Let us choose \( \mathbf{v} \) such that \( \mathbf{a}^T \mathbf{v} = 0 \) and \( \mathbf{c}^T \mathbf{v} \neq 0 \). Then, \( p(\mathbf{v}) = |\mathbf{a}^T \mathbf{v}| = 0 \) and

\[
p(\mathbf{A} \mathbf{v}) = |\mathbf{a}^T \mathbf{A} \mathbf{v}| = |\mathbf{a}^T \mathbf{u} \mathbf{c}^T \mathbf{v}| \neq 0.
\]

From Proposition 3.3.2(b) we conclude that \( p(A) = \infty \).

Thus we arrive at the following result.

Proposition 4.1.2. If \( p(\mathbf{v}) = |\mathbf{a}^T \mathbf{v}| \), then the threshold criterion holds if and only if one of the following two conditions holds:

(a) \( \mathbf{a}^T \mathbf{u} = 0 \) (in which case the adaptivity condition \( p(\mathbf{u}) > 0 \) is not satisfied);

(b) \( \mathbf{b}_0, \mathbf{b}_1, \mathbf{a} \) are collinear and \( |\alpha_0| \leq |\alpha_1| \).

Proof. If \( \mathbf{a}^T \mathbf{u} = 0 \), we conclude from Lemma 4.1.1 that \( p(\mathbf{A}_0) = p(\mathbf{A}_1^{-1}) = 1 \). This holds independently whether \( \mathbf{b}_0 \) and \( \mathbf{a} \) are collinear. Clearly, conditions in (4.2)-(4.3) are satisfied.

Consider now the case where \( \mathbf{a}^T \mathbf{u} \neq 0 \). If \( \mathbf{b}_0 \) and \( \mathbf{a} \) are collinear, the previous lemma yields that \( p(\mathbf{A}_0) = |\alpha_0| \) and \( p(\mathbf{A}_1^{-1}) = |\alpha_1|^{-1} \). Thus, from (4.2)-(4.3) we conclude that the threshold criterion holds if and only if \( |\alpha_0| \leq |\alpha_1| \).

If \( \mathbf{b}_0 \) and \( \mathbf{a} \) are not collinear, Lemma 4.1.1 yields that \( p(\mathbf{A}_0) = \infty \) and, consequently, the threshold criterion cannot hold.
Therefore, if the adaptivity condition on the weighted gradient seminorm is satisfied, the threshold criterion holds if and only if there exist constants $\gamma_0, \gamma_1 \in \mathbb{R}$ such that

$$|1 - \gamma_0 \sum_{j=1}^{N} a_j| \leq |1 - \gamma_1 \sum_{j=1}^{N} a_j| \quad \text{and} \quad \beta_{d,j} = \gamma_d a_j \quad \text{for } d = 0, 1 \text{ and } j = 1, \ldots, N.$$ 

**Example 4.1.3 (Choosing the update filter coefficients).** Consider the seminorm $p(v) = |a^T v|$ with $\sum_j a_j \neq 0$, where $\sum_j$ denotes the summation over all indices $j$. Following Proposition 4.1.2, we must choose $\beta_d = \gamma_d a$ such that

$$|1 - \gamma_0 \sum_j a_j| \leq |1 - \gamma_1 \sum_j a_j|.$$ 

The obvious question is how to choose the parameters $\gamma_1, \gamma_0$ such that the resulting filters have the 'right behavior'. What should be meant by 'right behavior' is, of course, strongly dependent on the goal of the filtering. In most practical cases, the updated signal $x'$ should be a coarser representation of the original $x_0$ where important features such as edges have been preserved (or perhaps even enhanced), while noise has been reduced and homogeneous regions have been simplified. Thus, we follow the premise of smoothing the signal to a certain degree (to reduce noise and avoid aliasing) but without blurring the edges (to preserve the most important visual features).

For example, for $d = 1$, in which case the gradient is large\(^1\), we may choose not to filter at all, i.e., $x' = x$ in (4.1). This can be achieved by choosing $\gamma_1 = 0$, which yields $\beta_1 = 0$; hence $\alpha_1 = 1$. In more homogeneous areas, where $d = 0$, we choose $\gamma_0$ in such a way that a low-pass filtering is performed. For instance, we may require that a given noise rejection criterion is maximized. If we assume that the input signal is contaminated by additive uncorrelated Gaussian noise, then it is easy to show\(^2\) that we must choose

$$\gamma_0 = \frac{\sum_j a_j}{\sum_j a_j^2 + (\sum_j a_j)^2} \quad \text{(4.5)}$$

for minimizing the variance of the noise in the approximation signal $x'$. This leads to $|\alpha_0| = (\sum_j a_j^2) / (\sum_j a_j^2 + (\sum_j a_j)^2) \leq 1$. Hence, if $|\alpha_1| \geq 1$, Proposition 4.1.2(b) is satisfied and we do have perfect reconstruction.

Obviously, an important parameter is the threshold $T$ which sets the frontier between 'high gradient' or 'edge' ($d = 1$) and 'homogeneous region' or 'non-edge' ($d = 0$). Thus, $T$ should be chosen carefully depending on the input signals, the seminorm, and the degree of 'edge-preservingness' one wishes to achieve.

\(^1\)Strictly speaking, $d = 1$ occurs when the seminorm of the gradient $v$ is above a given threshold $T$. For simplicity in our exposition, we say that the gradient is large when $d = 1$, and small when $d = 0$. We also make the implicit assumption that $d = 1$ corresponds to sharp transitions in the signal, while $d = 0$ corresponds to homogeneous or smooth regions.

\(^2\)Under such assumptions, the variance of the noise after the update lifting step is proportional to $(1 - \gamma_d \sum_j a_j)^2 + \gamma_d^2 \sum_j a_j^2$. 


4.1. Weighted gradient seminorm

Throughout the remainder of this subsection we deal with one-dimensional signals $x_0$ which are decomposed into two bands $x$ and $y$ (hence $P = 1$ in (3.1)). Furthermore, we consider $x(n) = x_0(2n)$ and $y(n) = x_0(2n + 1)$. So far, we have assumed that the gradient vector is indexed by $j = 1, \ldots, N$, that is, $v(n) = (v_1(n), \ldots, v_N(n))^T$. In this subsection, however, we assume that

$$v(n) = (v_{-K}(n), v_{-K+1}(n), \ldots, v_0(n), v_1(n), \ldots, v_{L-1}(n), v_L(n))^T$$

where

$$v_j(n) = x(n) - y(n + j) \quad \text{for } j = -K, \ldots, 0, \ldots, L.$$ 

An illustration is given in Fig. 4.1. With every coefficient vector $a \in \mathbb{R}^{K+L+1}$ in (4.4) we can associate a filter $\Delta_a$ which maps an input vector $(y(n-K), \ldots, y(n-1), x(n), y(n), \ldots, y(n+L))^T$, or equivalently, $(x_0(2n-2K+1), \ldots, x_0(2n-1), x_0(2n), x_0(2n+1), \ldots, x_0(2n+2L+1))^T$ onto an output value

$$\Delta_a(x_0)(2n) = \sum_{j=-K}^{L} a_j v_j(n) = \sum_{j=-K}^{L} a_j (x_0(2n) - x_0(2n + 2j + 1)) . \quad (4.6)$$

It is possible to choose the coefficients in such a way that it corresponds with an $N$'th-order discrete derivative filter for every $N$ with

$$N \leq L + K + 1.$$ 

For $N = 1$ and $K = L = 0$, the value $\Delta(x_0)(2n) = v_0(n) = x_0(2n) - x_0(2n + 1)$ is the first-order derivative. For $N = 2$ (with $K = 1$ and $L = 0$) and $a_{-1} = a_0 = 1$, we arrive at the expression:

$$\Delta(x_0)(2n) = v_0(n) + v_1(n) = 2x_0(2n) - x_0(2n - 1) - x_0(2n + 1),$$

which is a second-order derivative; see also Example 4.1.5 below.

We denote by $\mathcal{A}_N$, with $N \geq 1$, the coefficient vectors $a \in \mathbb{R}^{K+L+1}$ for which the corresponding filter $\Delta_a$ in (4.6) corresponds with an $N$'th-order derivative filter, or equivalently,
rejects signals that are polynomial of order less or equal than $N - 1$. The latter means that for all \( n \in \mathbb{Z} \),
\[
\sum_{j=\text{-}K}^{L} a_j \left[ (2n)^k - (2n + 2j + 1)^k \right] = 0 \quad \text{for} \quad k = 0, \ldots, N - 1,
\]
which is satisfied if and only if either $N = 1$, or $N > 1$ and
\[
\sum_{j=\text{-}K}^{L} a_j (2j + 1)^k = 0 \quad \text{for} \quad k = 1, \ldots, N - 1.
\]
Consider the function $Q_a$ given by
\[
Q_a(z) = \sum_{j=\text{-}K}^{L} a_j (1 - z^{2j+1}).
\]
The proof of the following result is straightforward.

**Lemma 4.1.4.** $a \in \mathcal{A}_N$ if and only if $Q_a$ has a zero at $z = 1$ with multiplicity $N$.

We next consider the case $N > 1$. Obviously, $Q_a$ has a zero of multiplicity $N$ if and only if $Q'_a$ (the derivative of $Q_a$ with respect to $z$) has a zero of multiplicity $N - 1$. Now
\[
Q'_a(z) = - \sum_{j=\text{-}K}^{L} a_j (2j + 1)z^{2j},
\]
and if $Q'_a$ has a zero at $z = 1$ with multiplicity $N - 1$, then we can write
\[
Q'_a(z) = (z - 1)^{N-1}z^{-2K}R(z),
\]
with $R(1) \neq 0$ and
\[
R(z) = \sum_{i=0}^{2(L+K)-N+1} r_i z^i.
\]
From the fact that $Q'_a$ is even (see (4.7)), we conclude that
\[
(z - 1)^{N-1}R(z) = (-1)^{N-1}(z + 1)^{N-1}R(-z).
\]
This yields that $R$ can be written as
\[
R(z) = (z + 1)^{N-1} \sum_{i=0}^{L+K+1-N} q_i z^{2i}.
\]
Substitution of (4.9) into (4.8) yields
\[ Q_a(z) = z^{-2K} (z^2 - 1)^{N-1} \sum_{i=0}^{L+K+1-N} q_i z^{2i} \]
\[ = z^{-2K} \sum_{l=0}^{N-1} \binom{N-1}{l} (-1)^{N-1-l} \sum_{i=0}^{L+K+1-N} q_i z^{2(i+l)}. \]

Recall that \( L + K + 1 \geq N > 1 \). Replacing the summation variable \( i \) by \( j = i + l \), we get
\[ Q_a(z) = z^{-2K} \sum_{j=0}^{L+K} \sum_{l=\max(0,j-L-K-1+N)}^{\min(N-1,N-l)} \binom{N-1}{l} (-1)^{N-1-l} q_{j-l} z^{2j}. \]
\[ = \sum_{j=-K}^{L} \sum_{l=\max(0,j-L-1+N)}^{\min(N-1,N-l+K)} \binom{N-1}{l} (-1)^{N-1-l} q_{j+l-K} z^{2j}. \]

In combination with (4.7), this yields the following expression for the coefficients \( a_j \):
\[ -(2j+1)a_j = \sum_{l=\max(0,j-L-1+N)}^{\min(N-1,N-l+K)} \binom{N-1}{l} (-1)^{N-1-l} q_{j+l-K} z^{2j}. \]

If \( N = L + K + 1 \), this expression reduces to
\[ -(2j+1)a_j = \binom{L+K}{j+K} (-1)^{L-j} q_0. \]

In particular, if \( K = L \) and \( N = 2L + 1 \) (odd-length filter), we get (setting \( q_0 = -1 \))
\[ a_j = \binom{2L}{2j+1} (-1)^{L+j} \left( \frac{2L}{L+j} \right), \tag{4.10} \]
and if \( K = L + 1 \) and \( N = 2L + 2 \) (even-length filter), we get
\[ a_j = \binom{2L+1}{2j+1} \frac{2L+1}{L+j+1}. \tag{4.11} \]

In the two previous cases, it can be shown that \( \sum_{j=-K}^{L} a_j \neq 0 \) and hence the adaptivity condition on the seminorm is satisfied. Indeed, in both cases this expression represents the sum of an alternating series whose terms have decreasing absolute values. As the first term is positive, the sum is nonzero.

Note that if \( a \in A_N \), then the corresponding decision map does not respond to polynomials up to degree \( N - 1 \), i.e., the corresponding expression \( |a^T v(n)| \) is zero for all \( n \). The use of this decision map allows us to smooth 'polynomial' regions of the signal which are distorted by low-amplitude noise, and to preserve transitions between such regions which are 'detected' by this decision map. In other words, a decision rule given by a \( N \)'th-order derivative operator is 'sensitive' to changes in signals of order less than or equal to \( N - 1 \).
Example 4.1.5 (2nd-order derivative). Consider the case where $K = 1$, $L = 0$ and $N = 2$. Then expression (4.11) yields $a = (a_1, a_0)^T = (1, 1)^T$. From Proposition 4.1.2 we conclude that the threshold criterion holds if $\beta_0 = \gamma_0(1,1)^T$ and $\beta_1 = \gamma_1(1,1)^T$ with

$$|1 - 2\gamma_0| \leq |1 - 2\gamma_1|.$$ 

Choosing $\gamma_1$ and $\gamma_0$ as in Example 4.1.3, we get $a_1 = 1$, $\beta_1 = 0$ and $a_0 = \frac{1}{3}$, $\beta_0 = \frac{1}{3}(1,1)^T$.

4.1.2 Simulations

In this subsection we show some simulation results using the weighted gradient seminorm $p(v) = |a^Tv|$. In all cases, we choose filter coefficients $a_d$, $\beta_d$ such that the threshold criterion holds. Note that given the weight vector $a$, once the parameters $\gamma_0$, $\gamma_1$ are chosen, the filters are determined.

The threshold $T$ is chosen rather heuristically, with its value depending on the test signal and the seminorm value $p(u)$.

For clarity of presentation, the decomposition signals and decision maps have been rescaled to the size of the original input signal. When displaying images, the gray values of the samples (pixels) have been scaled between 0 and 255 (histogram stretching).

We apply the adaptive schemes to one-dimensional (1D) signals as well as two-dimensional (2D) signals, i.e., images. In this latter case, we consider two different sampling schemes, namely, the quincunx and the square ($2 \times 2$) sampling schemes. The output images (approximation, detail and decision map) are shown at level 2 for the quincunx case and at level 1 for the square case. At those levels, the output images have been reduced by a factor of two both in the horizontal and in the vertical direction. However, as mentioned above, we rescale them to the original input image size for displaying purposes.

1D case

We consider, as in the last subsection, $x(n) = x_0(2n)$, $y(n) = x_0(2n+1)$, and a gradient vector $v(n)$ with components $v_j(n) = x(n) - y(n+j)$, $j = -K, \ldots, L$; see Fig. 4.1. We give two examples. In both cases, we choose the update filters following the criteria proposed in Example 4.1.3. We consider $N = 4$, with $K = 2$ and $L = 1$. After the update lifting step, a fixed prediction step of the form:

$$y'(n) = y(n) - \frac{1}{2}(x'(n) + x'(n+1)) \quad (4.12)$$

is applied. The overall scheme can be iterated over the approximation signal yielding an adaptive multiresolution decomposition.

Experiment 4.1.1 (Seminorm $p(v) = |u^Tv|$ for 1D, $N = 4$ - Fig. 4.2)

First we consider the case where $a = u = (1,1,1,1)^T$. Proposition 4.1.2 yields that we must choose $\beta_d = \gamma_d(1,1,1,1)^T$ for some constants $\gamma_0$, $\gamma_1$ such that

$$|1 - 4\gamma_0| \leq |1 - 4\gamma_1|.$$
For $d = 1$, we choose $\gamma_1 = 0$ and thus $\beta_1 = 0$. For $d = 0$ we choose $\gamma_0$ as in (4.5):
\[
\gamma_0 = 1/5 \text{ and hence } \beta_{0,j} = 1/5 \text{ for all } j.
\]
Note that since $\alpha_0 = 1 - \sum_{j=-2}^{1} 1/5 = 1/5$, for low-gradient regions where $d = 0$ the approximation value $x'(n)$ is computed by averaging the samples $y(n-2), y(n-1), x(n), y(n), y(n+1)$. In other words, the equivalent analysis low-pass filter is an average filter. For high-gradient regions where $d = 1$, $x'(n) = x(n)$, i.e., the equivalent filter is the identity filter.

The input signal (a fragment of the 'leleccum' signal from the wavelet toolbox in Matlab) is shown in Fig. 4.2(a). The approximation and the detail signals are depicted in Fig. 4.2(b) and (c), respectively, for the first, second and third level of the decomposition. These levels are displayed from bottom to top in each subfigure. A threshold of $T = 18$ has been used. The vertical dotted lines in Fig. 4.2(b) represent the locations where the decision map returns $d = 1$. For comparison, the decompositions obtained for both non-adaptive cases corresponding with fixed $d = 0$ and $d = 1$ are shown in Fig. 4.2(d)-(e) and Fig. 4.2(f)-(g), respectively.

Observe that the adaptive scheme tunes itself to the local structure of the signal: it yields a smoothed approximation signal except at locations where the gradient is large (i.e., $d = 1$). 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 shows only a single peak, avoiding the oscillatory behavior one encounters in the non-adaptive case with fixed $d = 0$. This oscillatory behavior can be noticed by carefully inspecting the details at the finest resolution level.

**Experiment 4.1.2 (Third order derivative seminorm for 1D - Fig. 4.3)**
Next, we choose $a$ such that the decision map does not respond to polynomial regions of order 3. This gives $a = (-1/3, 3, 3, -1/3)^T$. Choosing $\gamma_1 = 0$ and $\gamma_0$ as in (4.5), we get
\[
\gamma_0 = 4/35 \text{ and hence } \beta_0 = (4/35)a.
\]
We consider the input signal depicted in Fig. 4.3(a). This signal contains constant, linear and quadratic parts, plus uncorrelated Gaussian noise with variance 0.01. Figures 4.3(b)-(d) show the approximation signal at three subsequent levels of decomposition using a threshold $T = 3.4$. As before, the vertical dotted lines show the locations where the decision map equals $d = 1$. Again, we can observe that the adaptive scheme smooths the homogeneous regions but does not introduce intermediate points during sharp transitions. This allows removal of the noise while keeping the edges unaffected even at coarser scales.

**2D case: quincunx sampling scheme**
First we consider 2D signals that are decomposed into two bands $x$ and $y$ corresponding to the polyphase decomposition in a quincunx sampling scheme. Here, the signals $x$ and $y$ are defined

3Recall that higher levels correspond to coarser approximation and detail signals, and that the wavelet representation of the original signal is given by the coarsest approximation signal along with all the detail signals.
Figure 4.2: Decompositions (at levels 1, 2 and 3) corresponding with Experiment 4.1.1. (a) Original signal; (b)-(c) approximation and detail signals in the adaptive case using a threshold $T = 18$; (d)-(e) approximation and detail signals in the non-adaptive case with $d = 0$; (f)-(g) approximation and detail signals in the non-adaptive case with $d = 1$. 
4.1. Weighted gradient seminorm

Figure 4.3: Adaptive decomposition with polynomial criterion of order 3 corresponding with Experiment 4.1.2. (a) Original signal; (b)-(d) approximation signals at levels 1, 2 and 3 using a threshold $T = 3.4$. The vertical dotted lines show the locations where the decision map equals 1.

at all points $n = (m, n)^T$ with $m + n$ even and odd respectively. We use the labeling shown in Fig. 4.4 (where the argument $n$ has been omitted). The adaptive update lifting step is followed by a fixed prediction step of the form:

$$y'(m, n) = y(m, n) - \frac{1}{4} \sum_{j=1}^{4} x_j(m, n),$$

(4.13)

where $m + n$ is odd and $x_j(m, n)$, $j = 1, \ldots, 4$, are the four horizontal and vertical (updated) neighbors of $y(m, n)$. Repeated application of this scheme with respect to the approximation image yields an adaptive multiresolution decomposition.

Experiment 4.1.3 (Laplacian derivative seminorm for 2D quincunx - Fig. 4.5)
Consider the case where $p$ models the Laplacian operator, that is,

$$p(v) = \left| \sum_{j=1}^{4} v_j \right|.$$
Chapter 4. Adaptive update lifting: specific cases

Figure 4.4: Indexing of samples for a quincunx sampling structure centered at sample $x$.

In this case, Proposition 4.1.2 amounts to $\beta_{d,j} = \gamma_d$ for $j = 1, \ldots, 4$ and $|1 - 4\lambda_0| \leq |1 - 4\gamma_1|$. By choosing $\alpha_1 < 1$, we ensure that, in any case, a low-pass filtering is performed, albeit with a varying degree of smoothness depending on the decision $d$. We take $\gamma_0 = 1/5$ and $\gamma_1 = 1/20$.

We consider as input image the synthetic image shown at the top left of Fig. 4.5, and compute two levels of decomposition using a threshold $T = 20$. The decision map associated with level 2 is depicted at the top right. The black and white regions correspond to $d = 0$ and $d = 1$, respectively. Thus, the decision map displayed here shows the high-gradient regions (i.e., $d = 1$) of the approximation image at level 1 (not shown). The corresponding approximation and detail images are depicted in the middle row. For comparison, the decomposition images obtained in the non-adaptive case with fixed $d = 0$ are shown in the bottom row. One can appreciate that in the adaptive case the edges are not smoothed to the same extent as in the non-adaptive case.

2D case: square sampling scheme

Next, we consider a 2D decomposition with 4 bands corresponding with a square sampling structure as depicted in Fig. 4.6. Observe that this decomposition has the same structure as the one in Fig. 3.3. However, we have adopted a new notation $y_v, y_h, y_d$ of the $y$-bands, replacing $y(-1), y(-2), y(-3)$. This reflects the fact that, after the prediction stage, the corresponding outputs $y_v', y_h', y_d'$ are sometimes called the vertical, the horizontal, and the diagonal detail bands, respectively.

The input images $x, y_v, y_h, y_d$ are obtained by a polyphase decomposition of an original image $x_0$, that is: $x(m, n) = x_0(2m, 2n)$, $y_v(m, n) = x_0(2m, 2n + 1)$, $y_h(m, n) = x_0(2m + 1, 2n)$, $y_d(m, n) = x_0(2m + 1, 2n + 1)$. We label the eight samples surrounding $x(m, n)$ by $y_j(m, n)$, $j = 1, \ldots, 8$; see also Fig. 4.6.

In the experiment below, we compute the detail signals $y_v', y_h', y_d'$ with a prediction scheme
4.1. Weighted gradient seminorm

Figure 4.5: Decompositions (at level 2) corresponding with Experiment 4.1.3. Top: input image (left) and decision map (right) using a threshold $T = 20$. Middle: approximation (left) and detail (right) images in the adaptive case. Bottom: approximation (left) and detail (right) images in the non-adaptive case with $d = 0$.

Figure 4.6: Indexing of samples in a $3 \times 3$ window centered at $x_0(2m, 2n)$. 
as depicted in Fig. 4.7, with \( P_h(x') = P_v(x') = x' \) and \( P_d(x', y'_h, y'_v) = x' + y'_h + y'_v \). This yields
\[
\begin{align*}
y'_h &= y_h - x' \\
y'_v &= y_v - x' \\
y'_d &= y_d - x' - y'_v - y'_h.
\end{align*}
\]
Alternatively, \( y'_d = y_d + x' - y'_v - y'_h \). Note that the resulting 2D wavelet decomposition is non-separable.

Figure 4.7: 2D wavelet decomposition comprising an adaptive update lifting step (left) and three consecutive (fixed) prediction lifting steps (right).

Experiment 4.1.4 (Seminorm \( p(v) = |a^T v| \) for 2D square, \( N = 8 \) - Fig. 4.8)
Consider the seminorm given by
\[
p(v) = |a^T v| = \left| \sum_{j=1}^{8} a_j v_j \right|
\]
where \( a_1 + a_2 + \cdots + a_8 \neq 0 \). Note that this last condition guarantees the adaptivity condition for the seminorm. Recall that this condition is necessary for the scheme to be truly adaptive: if it is not satisfied, then \( p(v) \) does not depend on \( x \). According to Proposition 4.1.2, we choose the filter coefficients
\[
\beta_d = \gamma_d a \quad \text{with} \quad |1 - \gamma_0 \sum_{j=1}^{8} a_j| \leq |1 - \gamma_1 \sum_{j=1}^{8} a_j|.
\]
We have seen that for the 1D case, one can choose the coefficients \( a_j \) in such a way that the decision map 'ignores' polynomials up to a given degree; the seminorm \( p(v) \) corresponds with a derivative filter in this case. It is easy to see that this can be extended to 2D images. For example, the expression \(|x - y_h - y_v + y_d|\) corresponds with a first-order derivative with respect to
both horizontal and vertical directions. To obtain this expression, one must choose \( a_1 = a_4 = 1, a_8 = -1 \) and \( a_i = 0 \) for the other coefficients. For the second-order derivative (with respect to both directions) we can take \( a_1 = a_2 = a_3 = a_4 = 1 \) and \( a_5 = a_6 = a_7 = a_8 = -1/2 \). In this case \( \sum_{j=1}^{8} a_j = 2 \) and we must choose \( \gamma_0, \gamma_1 \) such that \( |1 - 2\gamma_0| \leq |1 - 2\gamma_1| \). In this experiment we consider this latter choice of \( a \), i.e., \( a = (1, 1, 1, 1, -1/2, -1/2, -1/2, -1/2)^T \), and we use \( \gamma_0 = 1/4 \) and \( \gamma_1 = 0 \). This means that for smooth regions where \( d = 0 \) we compute \( x' \) as a weighted average of \( x \) and its eight horizontal, vertical and diagonal neighbors, whereas for less homogeneous regions where \( d = 1 \) we do not perform any filtering, i.e., \( x' = x \).

As input image we take the synthetic image depicted at the top left of Fig. 4.8. In the second row we show the approximation and the horizontal detail images, at level 1, using a threshold \( T = 10 \). The corresponding decomposition images obtained in the non-adaptive case with fixed \( d = 0 \) are shown in the bottom row. Note that the adaptive scheme yields an approximation which preserves well the edges, and a detail image with less oscillatory effects than its non-adaptive counterpart. Note also from the decision map that the filter \( \Delta_a \) does not 'see' horizontal and vertical edges. Such edges are well preserved in the adaptive as well as in the non-adaptive case.

### 4.2 Quadratic seminorm

#### 4.2.1 Perfect reconstruction conditions

In this section we consider the case where \( p \) is a quadratic seminorm of the form:

\[
p(v) = (v^T M v)^{1/2}, \quad v \in \mathbb{R}^N,
\]

where \( M \) is a \( N \times N \) symmetric positive semi-definite matrix. Before we treat this general case, we deal with the classical \( l^2 \)-norm, also called the Euclidean norm. Thus \( M = I \), where \( I \) is the \( N \times N \) identity matrix. Note that in this case \( p(u) = \| u \|_{l^2} \), hence the adaptivity condition for the Euclidean norm is satisfied. We start with the following auxiliary result.

**Lemma 4.2.1.** Let \( p_2 \) be the quadratic norm given by \( p_2(v) = \| v \| = (v_1^2 + \cdots + v_N^2)^{1/2} \) and let \( A \) be the matrix \( A = I - u\beta^T \), where \( u, \beta \in \mathbb{R}^N \).

(a) If \( u, \beta \) are collinear, then \( p_2(A) = \| A \| = \max \{ 1, |\alpha| \} \), where \( \alpha = 1 - \beta^T u \).

(b) If \( u, \beta \) are not collinear, then \( p_2(A) > 1 \).

**Proof.** (a) If \( u, \beta \) are collinear, i.e., \( \beta = \mu u \) for some constant \( \mu \in \mathbb{R} \), then the matrix \( A = I - \mu uu^T \) is symmetric and we get that \( p_2(A) = \| A \| \) is the maximum absolute value of its eigenvalues [4]. According to Lemma 3.5.4, these eigenvalues are 1 and \( \alpha \). Thus \( p_2(A) = \max \{ 1, |\alpha| \} \).

(b) If \( u, \beta \) are not collinear, then we can decompose \( \beta \) as \( \beta = \mu u + c \) where \( c \neq 0 \) is orthogonal to \( u \). Now

\[
Ac = (I - u\beta^T)c = c - u(\mu u + c)^T c = c - (c^T c)u = c - \| c \|^2 u,
\]
Figure 4.8: Decompositions (at level 1) corresponding with Experiment 4.1.4. Top: input image (left) and decision map (right) using a threshold $T = 10$. Middle: approximation (left) and horizontal detail (right) images in the adaptive case. Bottom: approximation (left) and horizontal detail (right) images in the non-adaptive case with $d = 0$. 
whence we get that \[\|Ac\|_2^2 = \|c\|_2^2 + N\|c\|_4^4,\] where we have used that \[\|u\|_2^2 = N.\] Therefore

\[p_2(A) \geq \frac{\|Ac\|_2}{\|c\|} = (1 + N\|c\|_2^2)^{\frac{1}{2}} > 1,\]

which concludes the proof. \(\square\)

**Proposition 4.2.2.** Let \(p = p_2\) be the Euclidean norm. Then the threshold criterion holds if and only if \(u, \beta_0, \beta_1\) are collinear and \(|\alpha_0| \leq 1 \leq |\alpha_1|\).

**Proof.** We have \(A_0 = I - u\beta_0^T\) and \(A_1^{-1} = I - u\beta_1^T\) where \(\beta_1 = -\alpha_1^{-1}\beta_1\). From the previous lemma we infer that \(p(A_0) \geq 1\) and \(p(A_1^{-1}) \geq 1\). Now (4.2)-(4.3) yield that the threshold criterion holds if and only if \(p(A_0) = p(A_1^{-1}) = 1\). First, this requires that \(u, \beta_0, \beta_1\) are collinear. Then \(p(A_0) = \max\{1, |\alpha_0|\}\) and \(p(A_1^{-1}) = \max\{1, |\alpha_1|^{-1}\}\); here we have used that \(1 - \beta_1^T u = 1 + \alpha_1^{-1}(1 - \alpha_1) = \alpha_1^{-1}\). Reminding that \(\alpha_d \neq 0\) for \(d = 0, 1\), we obtain that the threshold criterion holds only if \(|\alpha_0| \leq 1 \leq |\alpha_1|\). This proves the result. \(\square\)

Now we are ready to consider the more general case in (4.17) with \(M\) an arbitrary \(N \times N\) symmetric positive semi-definite matrix. Thus, \(M\) can be decomposed as

\[M = Q\Lambda Q^T,\]  

where \(Q\) is an orthogonal matrix (i.e., \(Q^TQ = QQ^T = I\)) and \(\Lambda\) is a diagonal matrix with nonnegative entries, the eigenvalues of \(M\). The columns of \(Q\) are the (orthogonal) eigenvectors of \(M\). Define \(n\) as

\[n = \text{rank}(M) = \text{rank}(\Lambda) \leq N.\]

Without loss of generality we can assume that

\[\Lambda = \begin{pmatrix} \Lambda_{11} & 0 \\ 0 & 0 \end{pmatrix},\]  

where \(\Lambda_{11}\) is an \(n \times n\) diagonal matrix with strictly positive entries. Note that \(\Lambda_{11} = \Lambda\) if and only if \(n = N\). The corresponding decomposition of \(Q\) is given by

\[Q = (Q_1 \; Q_2),\]  

where \(Q_1, \; Q_2\) are \(N \times n\) and \(N \times (N - n)\) matrices, respectively (when \(n = N\), we shall adopt the conventions: \(Q = Q_1\) and \(Q_2 = 0\)). Here the columns of \(Q_1\) are the eigenvectors of \(M\) corresponding to the positive eigenvalues contained in \(\Lambda_{11}\). Observe that, instead of (4.18), we can also write

\[M = Q_1\Lambda_{11}Q_1^T.\]

The \(N \times n\) matrix \(Q_1\) is semi-orthogonal in the sense that \(Q_1^TQ_1 = I\).

After these mathematical preparations, we formulate our results concerning the quadratic seminorm of an \(N \times N\) matrix \(A\).
Lemma 4.2.3. Let $p$ be the quadratic seminorm given by (4.17), let $M$ be decomposed as in (4.18) and let $A$ be an $N \times N$ matrix. Then,

$$p(A) = \begin{cases} \|A_{11}^{-\frac{1}{2}} Q_1^T A Q_1 A_{11}^{-\frac{1}{2}}\| & \text{if } Q_1^T A Q_2 = 0 \\ \infty & \text{otherwise}, \end{cases} \tag{4.21}$$

where $\| \cdot \|$ is the standard Euclidean norm and $A_{11}, Q_1, Q_2$ are defined as in (4.19)-(4.20).

In particular, if $\text{rank}(M) = \text{rank}(A) = N$, then

$$p(A) = \|A_{11}^{-\frac{1}{2}} Q_1^T A Q_1 A_{11}^{-\frac{1}{2}}\|. \tag{4.22}$$

Proof. To compute $p(A)$ we have to maximize $(v^T A^T M A v)^{\frac{1}{2}}$ under the constraint $v^T M v = 1$. Substituting $w = Q^T v$, this amounts to maximizing $(w^T Q_1^T A Q_1 Q^T A Q_2 w)^{\frac{1}{2}}$ under the constraint $w^T A w = 1$. Define the matrix $B = Q^T A Q$; then

$$B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} = \begin{pmatrix} Q_1^T \\ Q_2^T \end{pmatrix} A \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} = \begin{pmatrix} Q_1^T A Q_1 \\ Q_2^T A Q_2 \end{pmatrix},$$

where $B_{11}$ is an $n \times n$ matrix. The expression we have to maximize is $(w^T B^T A B w)^{\frac{1}{2}}$. A simple computation shows that

$$B^T A B = \begin{pmatrix} B_{11}^T A_{11} B_{11} & B_{11}^T A_{11} B_{12} \\ B_{12}^T A_{11} B_{11} & B_{12}^T A_{11} B_{12} \end{pmatrix}.$$

Decomposing $w = (w_1 \ w_2)^T$, with $w_1 \in \mathbb{R}^n$ and $w_2 \in \mathbb{R}^{N-n}$, we get

$$w^T B^T A B w = w_1^T B_{11}^T A_{11} B_{11} w_1 + 2 w_1^T B_{11}^T A_{11} B_{12} w_2 + w_2^T B_{12}^T A_{11} B_{12} w_2. \tag{4.23}$$

Furthermore, the constraint $w^T A w = 1$ amounts to $w_1^T A_{11} w_1 = 1$. This constraint only involves $w_1$ and not $w_2$. This means that maximization of (4.23) yields $\infty$ unless $B_{12} = Q_1^T A Q_2 = 0$. This proves the second equality in (4.21).

Let us henceforth assume that $B_{12} = 0$. Thus

$$(p(A))^2 = \max \left\{ w_1^T B_{11}^T A_{11} B_{11} w_1 \mid w_1^T A_{11} w_1 = 1 \right\}$$

$$= \max \left\{ s^T A_{11}^{-\frac{1}{2}} B_{11}^T A_{11} A_{11}^{-\frac{1}{2}} B_{11} A_{11}^{-\frac{1}{2}} s \mid s^T s = 1 \right\}$$

$$= \max \left\{ \|A_{11}^{-\frac{1}{2}} B_{11} A_{11}^{-\frac{1}{2}} s\|^2 \mid \|s\|^2 = 1 \right\},$$

where we have substituted $s = A_{11}^{\frac{1}{2}} w_1$. This yields

$$p(A) = \|A_{11}^{-\frac{1}{2}} B_{11} A_{11}^{-\frac{1}{2}}\| = \|A_{11}^{\frac{1}{2}} Q_1^T A Q_1 A_{11}^{-\frac{1}{2}}\|,$$

which had to be proved.

Finally, if $\text{rank}(M) = N$ then $A_{11} = A$, $Q_1 = Q$ and $Q_2 = 0$, and thus (4.21) reduces to (4.22).
We apply this result to the matrix $A_d = I - u\beta_d^T$, $d = 0, 1$. Then

$$Q_1^TA_dQ_2 = Q_1^TQ_2 - Q_1^Tu(Q_2^T\beta_d)^T = -Q_1^Tu(Q_2^T\beta_d)^T,$$

since $Q_1^TQ_2 = 0$ by the orthogonality of $Q$. Therefore, $Q_1^TA_dQ_2 = 0$ if either (i) $Q_1^Tu = 0$ or (ii) $Q_2^T\beta_d = 0$. In case (i) we have $p(A_0) = p(A_1^{-1}) = \|Q_1^TQ_2\| = 1$ and the threshold criterion holds. Note, however, that we have $p(u) = 0$ and consequently the adaptivity condition on the seminorm does not hold in this case. We now consider case (ii) where $Q_2^T\beta_d = 0$. We compute $p(A_0)$ and $p(A_1^{-1})$ in this case:

$$p(A_0) = \|Q_1^T(I - u\beta_0^T)Q_1\| = \|I - \tilde{u}\beta_0^T\|,$$

where $\tilde{u} = \Lambda_{11}^{1/2}Q_1^Tu$ and $\beta_0 = \Lambda_{11}^{1/2}Q_1^T\beta_0$ are $n$-dimensional vectors. We conclude from Lemma 4.2.1 that $p(A_0) > 1$ if $\tilde{u}, \beta_0$ are not collinear and that $p(A_0) = \max\{1, |\tilde{\alpha}_0|\}$, with $\tilde{\alpha}_0 = 1 - \beta_0^T\tilde{u}$, if $\tilde{u}, \beta_0$ are collinear. Here we have assumed that $n > 1$ (the case $n = 1$ will be treated on page 78). Substitution of $\tilde{u}, \beta_0$ yields

$$\tilde{\alpha}_0 = 1 - u^TQ_1Q_1^T\beta_0.$$

A similar computation shows that $p(A_1^{-1}) > 1$ if $\tilde{u}, \beta_1$ are not collinear, where $\tilde{\beta}_1 = \Lambda_{11}^{-1/2}Q_1^T\beta_1$, and that $p(A_1^{-1}) = \max\{1, |\tilde{\alpha}_1|\}$ if $\tilde{u}, \beta_1$ are collinear. Here

$$\tilde{\alpha}_1 = (1 + \frac{1}{\alpha_1}u^TQ_1Q_1^T\beta_1)^{-1}.$$

**Lemma 4.2.4.** If $Q_1^Tu \neq 0$, the following two assertions are equivalent:

(i) $Mu, \beta_d$ are collinear

(ii) $\tilde{u}, \tilde{\beta}_d$ are collinear and $Q_1^T\beta_d = 0$.

Proof. Assume (i). We have $Mu \neq 0$ (otherwise $Q_1^TMu = \Lambda_{11}Q_1^Tu = 0$) and then $\beta_d = \mu_dMu = \mu_dQ_1\Lambda_{11}Q_1^Tu$, where $\mu_d \in \mathbb{R}$. Since $Q_1^TQ_2 = 0$, we find that $Q_1^T\beta_d = 0$. Furthermore, $\tilde{\beta}_d = \Lambda_{11}^{-1/2}Q_1^T\beta_d = \mu_d\Lambda_{11}^{1/2}Q_1^Tu = \mu_d\tilde{u}$, where we have used that $Q_1^TQ_1 = I$.

Assume (ii): $Q_1^T\beta_d = 0$ is equivalent to $\beta_d \in \text{Ran}(Q_2)^\perp = \text{Ran}(Q_1)$, i.e., $\beta_d = Q_1\xi_d$, where $\xi_d \in \mathbb{R}^n$. Since $\tilde{u}$ and $\tilde{\beta}_d$ are collinear, we have $\tilde{\beta}_d = \mu_d\tilde{u}$, that is $\Lambda_{11}^{1/2}Q_1^T\beta_d = \mu_d\Lambda_{11}^{1/2}Q_1^Tu$, which yields $\xi_d = \mu_d\Lambda_{11}Q_1^Tu$, and hence $\beta_d = \mu_dQ_1\Lambda_{11}Q_1^Tu = \mu_dMu$. This concludes the proof.

Therefore, if $\beta_d = \mu_dMu$ with $\mu_d \in \mathbb{R}$, we get

$$\tilde{\alpha}_0 = 1 - u^TQ_1Q_1^T\beta_0 = 1 - \mu_0u^TQ_1\Lambda_{11}Q_1^Tu = 1 - u^T\beta_0 = \alpha_0$$

and

$$\tilde{\alpha}_1 = (1 + \frac{1}{\alpha_1}u^TQ_1Q_1^T\beta_1)^{-1} = (1 + \frac{1}{\alpha_1}(1 - \alpha_1))^{-1} = \alpha_1;$$

hence $p(A_0) = \max\{1, |\alpha_0|\}$ and $p(A_1^{-1}) = \max\{1, |\alpha_1|^{-1}\}$. Thus, we arrive at the following result.
Proposition 4.2.5. Let $p$ be the quadratic seminorm given by (4.17), let $M$ be decomposed as in (4.18), and assume that $n = \text{rank}(M) \geq 2$. Then the threshold criterion holds if and only if any of the following two conditions holds:

(a) $Q^T u = 0$ (in which case the adaptivity condition $p(u) > 0$ is not satisfied);

(b) $\beta_0, \beta_1, M u$ are collinear and $|\alpha_0| \leq 1 \leq |\alpha_1|$.

If $n = 1$, then it follows that $M = aa^T$, where $a \in \mathbb{R}^N$, $a \neq 0$. In this case, $p(v) = (v^T M v)^{1/2} = |a^T v|$ for all $v \in \mathbb{R}^N$, which yields the weighted gradient seminorm studied in Section 4.1.

Observe that Proposition 4.2.2, where $p$ corresponds to the Euclidean norm, is only a special case of the last proposition. The following example illustrates two other cases.

Example 4.2.6. (a) Consider first the case where $M = \Lambda$ with $\Lambda$ a diagonal matrix with strictly positive entries $\Lambda_{jj} = \lambda_j$ for $j = 1, \ldots, N$. Note that in this case we can write $p(v) = (v^T M v)^{1/2}$ as

$$p(v) = \left( \sum_{j=1}^N \lambda_j v_j^2 \right)^{1/2},$$

which can be regarded as a (positive) weighted Euclidean norm. Obviously, if $\lambda_j = 1$ for all $j$, i.e., $M$ is the identity matrix, then we are back at the standard Euclidean norm. According to Proposition 4.2.5, the threshold criterion holds if and only if there are constants $\mu_0, \mu_1$ such that $\beta_{d,j} = \mu_d \lambda_j$ for $d = 0, 1$ and $j = 1, \ldots, N$, and

$$|1 - \mu_0 (\lambda_1 + \cdots + \lambda_N)| \leq 1 \leq |1 - \mu_1 (\lambda_1 + \cdots + \lambda_N)|. \tag{4.24}$$

If we assume that the input signal is contaminated by additive uncorrelated Gaussian noise, it is easy to show (as in (4.5)) that we must take

$$\mu_d = \frac{\sum_j \lambda_j}{\sum_j \lambda_j^2 + (\sum_j \lambda_j)^2} \tag{4.25}$$

for minimizing the variance of the noise in the approximation signal. Here $\sum_j$ denotes summation over all indices $j$. If we take $\mu_0$ as in (4.25), it is then obvious that the first inequality in condition (4.24) is satisfied. Choosing, for example, $\mu_1 = 0$, we do have perfect reconstruction.

(b) Consider the same case as in (a) but with $\lambda_1, \ldots, \lambda_n$ strictly positive and $\lambda_{n+1} = \cdots = \lambda_N = 0$. The threshold criterion requires that $\beta_d$ is collinear with $M u$. This means that $\beta_{d,n+1} = \cdots = \beta_d N = 0$. In other words, the order of the update filter, initially assumed to be equal to $N$, is only $n$, and we are now back in the situation described in (a).
4.2. Quadratic seminorm

4.2.2 Simulations

In this subsection we show some simulation results using the quadratic seminorm \( p(v) = (v^T M v)^{1/2} \). For simplicity, we consider the case where \( M \) is a diagonal matrix with strictly positive entries such as in Example 4.2.6(a). The remarks made at the beginning of Section 4.1.2 apply also here.

**1D case**

As in Section 4.1.2 for the 1D case, we assume \( x(n) = x_0(2n) \), \( y(n) = x_0(2n + 1) \) and a gradient vector \( v(n) \) indexed as in Fig. 4.1. As before, a fixed prediction of the form \( y'(n) = y(n) - \frac{1}{2}(x'(n) + x'(n + 1)) \) is applied after the update step.

**Experiment 4.2.1 (Quadratic seminorm for 1D, \( N = 4 \) - Fig. 4.9)**

We repeat experiment 4.1.1 but with the weighted Euclidean norm given by

\[
p(v) = \left( \sum_{j=-2}^{1} \lambda_j v_j^2 \right)^{1/2}
\]

with weights \( (\lambda_{-2}, \lambda_{-1}, \lambda_0, \lambda_1) = \left( \frac{1}{3}, 1, 1, \frac{1}{3} \right) \),

or equivalently, \( M = \text{diag}(\frac{1}{3}, 1, 1, \frac{1}{3}) \). Following Example 4.2.6(a), we see that the threshold criterion holds if \( \beta_d = \mu_d \left( \frac{1}{3}, 1, 1, \frac{1}{3} \right)^{1/2} \), with

\[
|1 - \frac{8}{3} \mu_0| \leq 1 \leq |1 - \frac{8}{3} \mu_1|.
\]

We take \( \mu_1 = 0 \) and compute \( \mu_0 \) from (4.25), which yields \( \mu_0 = 2/7 \).

Again, we can observe from Fig. 4.9 that the adaptive scheme tunes itself to the local structure of the signal: it 'recognizes' and preserves the discontinuities, while smoothing the more homogeneous regions. This results in a detail signal which is small in homogeneous regions. Near singularities, however, the detail signal comprises a single peak, thus avoiding the oscillatory behavior exhibited by the non-adaptive case with fixed \( d = 0 \).

**Experiment 4.2.2 (Quadratic seminorm for 1D, \( N = 6 \) - Fig. 4.10)**

Now we assume \( M = \text{diag}(\frac{1}{3}, \frac{1}{2}, 1, 1, \frac{1}{2}, \frac{1}{3}) \). As in the previous example, in order to satisfy the threshold criterion (and hence guarantee perfect reconstruction), we must take \( \beta_{d,j} = \mu_d \lambda_j \), \( j = -3, \ldots, 2 \), for \( d = 0, 1 \), and choose constants \( \mu_0, \mu_1 \) such that (4.24) holds. Here, we choose these constants such that the equivalent filter is a low-pass filter with \( \alpha_0 = \beta_{0,0} = \beta_{0,-1} \) for \( d = 0 \), and the identity filter for \( d = 1 \). More precisely, we choose

\[
\mu_0 = \left( 1 + \sum_{j=-3}^{2} \lambda_j \right)^{-1} \quad \text{and} \quad \mu_1 = 0.
\]

Fig. 4.10 shows the approximation signals at three subsequent levels of decomposition, for three different thresholds as well as for the non-adaptive scheme with fixed \( d = 0 \). By varying the
Figure 4.9: Decompositions (at levels 1, 2 and 3) corresponding with Experiment 4.2.1. (a) Original signal; (b)-(c) approximation and detail signals in the adaptive case using a threshold $T = 18$; (d)-(e) approximation and detail signals in the non-adaptive case with $d = 0$; (f)-(g) approximation and detail signals in the non-adaptive case with $d = 1$. 
threshold $T$, the adaptive system can be tuned to one of its non-adaptive counterparts (fixed $d = 0$ or $d = 1$). If $T$ is very small, the adaptive system will behave more or less as the non-adaptive scheme with fixed $d = 1$ (not shown). If $T$ is increased, the decision map will attain the value 0 more often, meaning that it behaves increasingly as the non-adaptive scheme with fixed $d = 0$ (Fig. 4.10(e)). Obviously, this general observation is valid for all the adaptive schemes described so far.

Figure 4.10: Decompositions (at levels 1, 2 and 3) corresponding with Experiment 4.2.2. (a) Original signal; (b)-(c) approximation in the adaptive case using a threshold $T = 15$ and $T = 20$; (d)-(e) approximation in the adaptive case using a threshold $T = 30$ and non-adaptive case with $d = 0$. 
2D case: quincunx sampling scheme

In the following two examples we use a quincunx decomposition as depicted in Fig. 4.4. As in Experiment 4.1.3 (see (4.13)), the prediction of each sample $y(m,n)$ is computed by averaging its four horizontal and vertical updated neighbors.

**Experiment 4.2.3 (Quadratic seminorm for 2D quincunx, $N = 4$ - Fig. 4.11)**

Consider the Euclidean norm and $N = 4$. Proposition 4.2.2 implies that $\beta_{d,j} = \beta_{d}$ for $j = 1, \ldots, 4$, and condition $|a_0| \leq 1 \leq |a_1|$ reduces to

$$|1 - 4\beta_0| \leq 1 \leq |1 - 4\beta_1|.$$ 

A possible solution is $\beta_0 = 1/5$ and $\beta_1 = 0$. This choice means that in homogeneous areas where $d = 0$, the approximation signal $x$ is averaged with its four neighbors whereas in the vicinity of singularities where $d = 1$, no filtering is performed.

As input image we choose the ‘House’ image shown at the top left of Fig. 4.11. We take a threshold $T = 60$. The approximation and detail images obtained after two levels of decomposition are shown in the middle row. The corresponding decision map is depicted at the top right of Fig. 4.11. The approximation and detail images obtained in the non-adaptive case with fixed $d = 0$ are shown in the bottom row.

**Experiment 4.2.4 (Quadratic seminorm for 2D quincunx, $N = 12$ - Fig. 4.12)**

In this experiment we choose update filters with a larger support, namely the samples labeled by $y_1, \ldots, y_{12}$ in Fig. 4.4.

We choose a quadratic norm like in Example 4.2.6(a) where $M_{jj} = \lambda_j$ is the inverse value of the distance of the corresponding sample to the center $x$. This leads to $M = \text{diag}(1,1,1,1,1/\sqrt{5}, \ldots, 1/\sqrt{5})$, and now the formula in (4.25) yields $\rho_0 = \frac{5+2\sqrt{5}}{43+16\sqrt{5}}$. Choosing $\rho_1 = 0$, we have

$$\beta_0 = \rho_0(1,1,1,1,\frac{1}{\sqrt{5}}, \ldots, \frac{1}{\sqrt{5}})^T \quad \text{and} \quad \beta_1 = 0.$$ 

As before, the input image is the ‘House’ image depicted at the top left of Fig. 4.12. We take a threshold $T = 82.5$. The approximation and detail images, after two levels of decomposition, are shown in the middle row. The corresponding decision map is depicted at the top right. The decomposition images in the non-adaptive case with fixed $d = 0$ are shown in the bottom row. Again, we can observe that in the adaptive case the edges are better preserved than in the non-adaptive case. We note that the improvement is more visible than in the previous experiment, which is partly due to the fact that the filter length is larger.

2D case: square sampling scheme

In the next experiment we consider a 2D decomposition with 4 bands as depicted in Fig. 4.6. As in Experiment 4.1.4, we consider the lifting scheme shown in Fig. 4.7 with the prediction filters given by (4.14)-(4.16).
Figure 4.11: Decompositions (at level 2) corresponding with Experiment 4.2.3. Top: input image (left) and decision map (right) using a threshold $T = 60$. Middle: approximation (left) and detail (right) images in the adaptive case. Bottom: approximation (left) and detail (right) images in the non-adaptive case with $d = 0$. 
Figure 4.12: Decompositions (at level 2) corresponding with Experiment 4.2.4. Top: input image (left) and decision map (right) using a threshold $T = 82.5$. Middle: approximation (left) and detail (right) images in the adaptive case. Bottom: approximation (left) and detail (right) images in the non-adaptive case with $d = 0$. 
Experiment 4.2.5 (Quadratic seminorm for 2D square, \(N = 8\) - Fig. 4.13)
Here we consider the seminorm
\[
p(v) = \left( \sum_{j=1}^{4} |v_j|^2 + \frac{1}{2} \left( \sum_{j=5}^{8} |v_j|^2 \right) \right)^{1/2}.
\]
This corresponds with Example 4.2.6(a) where \(M = \text{diag}(1, 1, 1, 1, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})\). Thus the threshold criterion holds if we choose \(\beta_d = \mu_d (1, 1, 1, 1, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})^T\). As before, we take \(\mu_1 = 0\) and compute \(\mu_0\) from (4.25), which gives \(\mu_0 = 6/41\).

The input image is the ‘Trui’ image shown at the top left of Fig. 4.13. We use a threshold \(T = 49\). The approximation and the horizontal detail images, after one level of decomposition, are depicted in the middle row, and the decision map at the top right. The corresponding decomposition images for the non-adaptive case with fixed \(d = 0\) are shown at the bottom row.

4.3 \(l^1\)-norm and \(l^\infty\)-norm

4.3.1 Perfect reconstruction conditions

The following result, which applies to the situation where \(p\) is a norm rather than a seminorm, is straightforward.

**Observation 4.3.1.** Let \(p\) be a norm and \(A\) a bounded linear operator. Then, the adaptivity condition \(p(u) > 0\) is satisfied. Furthermore, \(p(A) < \infty\).

In this section we concentrate on the case where \(p\) is the \(l^1\)-norm

\[
p_1(v) = \sum_{j=1}^{N} |v_j|,
\]
or the \(l^\infty\)-norm

\[
p_\infty(v) = \max_{j=1,...,N} |v_j|.
\]

Recall that \(A_d = I - u\beta_d^T\) and that \(\alpha_d = \det(A_d) \neq 0\). In addition, we assume that \(N > 1\).

**Proposition 4.3.2.** If \(p = p_1\), then the threshold criterion holds if and only if \(N = 2, \beta_{0,1}, \beta_{0,2} \in [0, 1]\) and either \(\beta_{1,1}, \beta_{1,2} \leq 0\) or \(\beta_{1,1}, \beta_{1,2} \geq 1\).

*Proof.* From the above observation we have that \(p(A_0) < \infty\) and \(p(A_1^{-1}) < \infty\). Thus, (4.2)-(4.3) reduce to \(p(A_0)p(A_1^{-1}) \leq 1\). The \(l^1\)-norm of the matrix \(A_d\) is given [4] by

\[
p_1(A_d) = \max_{j} (|1 - \beta_{d,j}| + (N - 1)|\beta_{d,j}|),
\]
Figure 4.13: Decompositions (at level 1) corresponding with Experiment 4.2.5. Top: original (left) and decision map (right) using a threshold $T = 49$. Middle: approximation (left) and horizontal detail (right) images in the adaptive case. Bottom: approximation (left) and horizontal detail (right) images in the non-adaptive case with $d = 0$. 
and the norm of its inverse is

\[ p_1(A_d^{-1}) = \max_j \left( |1 + \frac{\beta_{d,j}}{\alpha_d}| + (N - 1)\frac{|\beta_{d,j}|}{|\alpha_d|} \right). \]

Therefore, condition \( p_1(A_0)p_1(A_1^{-1}) \leq 1 \) becomes

\[
\max_j (|1 - \beta_{0,j}| + (N - 1)|\beta_{0,j}|) \cdot \max_j \left( |1 + \frac{\beta_{1,j}}{\alpha_1}| + (N - 1)\frac{|\beta_{1,j}|}{|\alpha_1|} \right) \leq 1.
\]

Recall that \( N \geq 2 \). Let us first observe that for any \( j = 1, \ldots, N \), we have

\[
|1 - \beta_{0,j}| + (N - 1)|\beta_{0,j}| = \begin{cases} 1 + N|\beta_{0,j}| > 1 & \text{if } \beta_{0,j} < 0 \\ 1 + (N - 2)|\beta_{0,j}| \geq 1 & \text{if } 0 \leq \beta_{0,j} \leq 1 \\ N|\beta_{0,j}| - 1 > 1 & \text{if } \beta_{0,j} > 1 \end{cases}
\]

and

\[
|1 + \frac{\beta_{1,j}}{\alpha_1}| + (N - 1)\frac{|\beta_{1,j}|}{|\alpha_1|} = \begin{cases} 1 + N\frac{|\beta_{1,j}|}{|\alpha_1|} > 1 & \text{if } \text{sign } \beta_{1,j} = \text{sign } \alpha_1 \\ 1 + (N - 2)\frac{|\beta_{1,j}|}{|\alpha_1|} \geq 1 & \text{if } \text{sign } \beta_{1,j} \neq \text{sign } \alpha_1 \text{ and } |\alpha_1| \geq |\beta_{1,j}| \end{cases}
\]

Thus, \( p_1(A_0) \geq 1 \) and \( p_1(A_1^{-1}) \geq 1 \). Consequently, condition \( p_1(A_0)p_1(A_1^{-1}) \leq 1 \) can only be satisfied when \( p_1(A_0) = p_1(A_1^{-1}) = 1 \). The equality \( p_1(A_0) = 1 \) implies that for any \( j = 1, \ldots, N \), either \( \beta_{0,j} = 0 \) or \( N = 2 \) and \( 0 \leq \beta_{0,j} \leq 1 \). The equality \( p_1(A_1^{-1}) = 1 \) means that for any \( j = 1, \ldots, N \), either \( \beta_{1,j} = 0 \) or \( N = 2 \), \( \text{sign } \beta_{1,j} \neq \text{sign } \alpha_1 \) and \( |\alpha_1| \geq |\beta_{1,j}| \). From these implications, Proposition 4.3.2 follows immediately. \( \square \)

Next, we consider \( p \) to be the \( l^{\infty} \)-norm. We will see that in this case the conditions on the filter coefficients are slightly more restrictive than in the previous case.

**Proposition 4.3.3.** Assume \( p = p_\infty \), then the threshold criterion holds if and only if \( N = 2 \), \( \beta_{0,1} = \beta_{0,2} \in [0,1] \) and either \( \beta_{1,1} = \beta_{1,2} \leq 0 \) or \( \beta_{1,1} = \beta_{1,2} \geq 1 \).

**Proof.** Again, (4.2)-(4.3) reduce to \( p(A_0)p(A_1^{-1}) \leq 1 \). The \( l^{\infty} \)-norm of the matrix \( A_d \) is given \([4]\) by

\[ p_\infty(A_d) = \max_i \left( |1 - \beta_{d,i}| + \sum_{j \neq i} |\beta_{d,j}| \right), \]

and the norm of its inverse is

\[ p_\infty(A_d^{-1}) = \max_i \left( |1 + \frac{\beta_{d,i}}{\alpha_d}| + \sum_{j \neq i} \frac{|\beta_{d,j}|}{|\alpha_d|} \right). \]
Chapter 4. Adaptive update lifting: specific cases

Recall that $N \geq 2$. The $l^\infty$-norm of $A_0$ can be expressed as

$$p_\infty(A_0) = \begin{cases} 1 + \sum_j |\beta_{0,j}| & \text{if } \beta_{0,j} < 0 \text{ for some } j = 1, \ldots, N \\ |1 - \beta_{0,m}| + \sum_{j \neq m} |\beta_{0,j}| & \text{otherwise} \end{cases}$$

where $m = \text{argmin}_j \beta_{0,j}$. Likewise, the $l^\infty$-norm of $A_{1}^{-1}$ is

$$p_\infty(A_{1}^{-1}) = \begin{cases} 1 + \sum_j |\beta_{1,j}| & \text{if } \text{sign } \alpha_1 = \text{sign } \beta_{1,j} \text{ for some } j = 1, \ldots, N \\ |1 + \frac{\beta_{1,m}}{\alpha_1}| + \sum_{j \neq m} |\frac{\beta_{1,j}}{\alpha_1}| & \text{if } \beta_{1,j} = 0 \text{ for all } j = 1, \ldots, N \\ > 1 & \text{otherwise} \end{cases}$$

where $m = \text{argmin}_j \beta_{1,j}$. Thus, both $p_\infty(A_0)$ and $p_\infty(A_{1}^{-1})$ values are at least 1, which means that condition $p_\infty(A_0)p_\infty(A_{1}^{-1}) \leq 1$ holds only if $p_\infty(A_0) = p_\infty(A_{1}^{-1}) = 1$, which in turn is satisfied only under the conditions stated in the proposition. $\square$

4.3.2 Simulations

In this subsection we show some simulation results using the $l^1$-norm and the $l^\infty$-norm. We only consider the 1D case with $x(n) = x_0(2n)$ and $y(n) = x_0(2n + 1)$. As in previous 1D simulations, the prediction step is of the form $y'(n) = y(n) - \{x'(n) + x'(n + 1)\}$.

Experiment 4.3.1 ($l^1$-norm and $l^\infty$-norm for 1D, $N = 2$ - Fig. 4.14)

Assuming $N > 1$, the threshold criterion can only be satisfied if $N = 2$. Thus, we consider the norms

$$p_1(v) = |v_1| + |v_2|$$
$$p_\infty(v) = \max\{|v_1|, |v_2|\}.$$ 

We choose $\beta_{0,1} = \beta_{0,2} = \frac{1}{3}$, and $\beta_{1,1} = \beta_{1,2} = 0$. Thus, the resulting low-pass filters are the average filter for $d = 0$, and the identity filter for $d = 1$. The original input signal $x_0$ is shown at the top left of Fig. 4.14. The approximation and detail signals, $x'$ and $y'$, are depicted in the second row for the $l^1$-norm, and in the third row for the $l^\infty$-norm. In both cases we have taken a threshold$^4$ $T = 0.28$. The locations where the decision maps return $d = 1$ are shown as vertical dotted lines in the corresponding approximation figures. Since $p_\infty(v) \leq p_1(v)$, if $d = 1$ for the $l^\infty$-norm, then $d = 1$ for the $l^1$-norm (but not vice versa). The decomposition signals obtained for both non-adaptive cases with $d = 0$ and $d = 1$ are shown respectively in the fourth and fifth rows of Fig. 4.14. As in previous experiments, the adaptive schemes smooth

$^4$If one would like both adaptive schemes to be comparable, it would be more appropriate to choose different thresholds for each norm, e.g., $T_1$ for $p = p_1$ and $T_\infty = T_1/2$ for $p = p_\infty$. 


the signal while preserving the sharp transitions detected by the corresponding decision maps. 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 observes in the non-adaptive case with fixed \( d = 0 \).

### 4.4 Continuous decision map

In the previous sections we have been dealing exclusively with binary decision maps \( D \) whose output \( d \in \{0, 1\} \) is obtained by thresholding the seminorm of the gradient vector \( v \in \mathbb{R}^N \), i.e.,

\[
d = [p(v) > T].
\]

In this section, we consider decision maps \( D \) whose output \( d \) can take values in a continuous interval.

#### 4.4.1 Perfect reconstruction conditions

Consider an update step of the form:

\[
x' = \alpha_d x + \beta_{d,1} y_1 + \beta_{d,2} y_2
\]

where \( d = D(v) \), \( D: \mathbb{R}^2 \to \mathcal{D} \). Here \( \mathcal{D} \subseteq \mathbb{R} \) is the decision set containing all possible decisions \( d \). Note that the decision depends on the gradient vector \( v = (v_1, v_2)^T \) but it is not restricted to have discrete values, and hence we have filter coefficients \( \alpha_d, \beta_{d,1}, \beta_{d,2} \in \mathbb{R} \) for every \( d \in \mathcal{D} \).

Using the same notation as in Chapter 3, we define

\[
\kappa_d = \alpha_d + \beta_{d,1} + \beta_{d,2}.
\]

**Lemma 4.4.1.** In order to have perfect reconstruction, it is necessary that \( \kappa_d \) is constant on every subset \( \mathcal{D}(c) \subseteq \mathcal{D} \) given by \( \mathcal{D}(c) = \{ D(v_1, v_2) \mid v_1 - v_2 = c \} \), where \( c \in \mathbb{R} \) is a constant.

**Proof.** Assume that, for some \( c \in \mathbb{R} \), we have \( a, b \in \mathcal{D}(c) \) such that \( \kappa_a \neq \kappa_b \). Assume also that \( v_d = (v_{d,1}, v_{d,2})^T \) is such that \( D(v_{d,1}, v_{d,2}) = d \) for \( d = a, b \). Choose inputs \( x = \xi + v_{a,1}, y_1 = \xi \) and \( y_2 = \xi + v_{a,1} - v_{a,2} = \xi + c \). From (4.26) we get

\[
x' = \alpha_a (\xi + v_{a,1}) + \beta_{a,1} \xi + \beta_{a,2} (\xi + c)
\]

\[
= \kappa_a (\xi + v_{a,1}) - (\beta_{a,1} + \beta_{a,2}) v_{a,1} + \beta_{a,2} c
\]

\[
= \kappa_a \xi + \kappa_a v_{a,1} - \beta_{a,1} v_{a,1} - \beta_{a,2} v_{a,2}.
\]

Now, if we take \( x = \xi + v_{b,1} \) and the same \( y_1, y_2 \) as before, we get

\[
x' = \kappa_b \xi + \kappa_b v_{b,1} - \beta_{b,1} v_{b,1} - \beta_{b,2} v_{b,2}.
\]
Figure 4.14: Decompositions (at level 1) corresponding with Experiment 4.3.1. Top: original signal. Second and third rows: approximation (left) and detail (right) signals for the $l^1$-norm and the $l^\infty$-norm, respectively, using a threshold $T = 0.28$. Fourth and bottom rows: approximation (left) and detail (right) signals in the non-adaptive cases with $d = 0$ and $d = 1$, respectively.
Choose $\xi$ in such a way that

$$\kappa_a \xi + \kappa_a v_{a,1} - \beta_{b,1} v_{a,1} - \beta_{b,2} v_{a,2} = \kappa_b \xi + \kappa_b v_{b,1} - \beta_{b,1} v_{b,1} - \beta_{b,2} v_{b,2},$$

which is possible since $\kappa_a \neq \kappa_b$. Thus, we get that for the same values of $y_1, y_2$, two different inputs for $x$ may yield the same output. This implies that perfect reconstruction is not possible.

Moreover, if the decision map is of the form:

$$d = D(|v_1| + |v_2|),$$

(4.27)

for $D: \mathbb{R}_+ \rightarrow \mathcal{D}$, then $\mathcal{D}(0) = \mathcal{D}$. Thus we arrive at the following result.

**Lemma 4.4.2.** Assume that the decision map is given by (4.27). In order to have perfect reconstruction it is necessary that $\kappa_d$ does not depend on $d$.

As we did for the binary decision map, we assume $\kappa_d = 1$ and $\alpha_d \neq 0$ for all $d \in \mathcal{D}$. It is straightforward that

$$v' = A_d v \quad \text{with} \quad A_d = \begin{pmatrix} 1 - \beta_{d,1} & -\beta_{d,2} \\ -\beta_{d,1} & 1 - \beta_{d,2} \end{pmatrix}.$$

Since $A_d$ is invertible, we can recover $v$ from $v'$ assuming we know the coefficients $\alpha_d, \beta_{d,1}, \beta_{d,2}$ which all depend on $d = D(|v_1| + |v_2|)$. This leads 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 $v = (v_1, v_2)^T \in \mathbb{R}^2$.

Henceforth, we analyze the particular case where the decision $d$ equals the $l^1$-norm of the gradient, i.e.,

$$d = |x - y_1| + |x - y_2| = |v_1| + |v_2|,$$

(4.28)

corresponding with a possibly infinite collection of update filters parameterized by $d$.

**Proposition 4.4.3.** Assume an update step as in (4.26) where the decision $d$ is given by (4.28). Perfect reconstruction is possible in each of the following two cases:

(a) $\alpha_d > 0$ for all $d \geq 0$, and $\beta_{d,1}, \beta_{d,2}$ are non-increasing with respect to $d$.

(b) $\alpha_d < 0$ for all $d \geq 0$, and $\beta_{d,1}, \beta_{d,2}$ are non-decreasing with respect to $d$.

**Proof.** Consider an input sample $x_k$ whose update is given by

$$x_k' = \alpha_d x_k + \beta_{d,1} y_1 + \beta_{d,2} y_2,$$

and whose corresponding gradient vector is $v_k = (x_k - y_1, x_k - y_2)^T$. Assume $d_1, d_2 \in \mathcal{D}$. We show that $x_1 \neq x_2$ implies that $x_1' \neq x_2'$ in both cases (a) and (b) of the above proposition. Without loss of generality we may assume $y_2 \geq y_1$ and $x_2 > x_1$. For simplicity in the expressions,
we introduce the following notation for the coefficients in $\beta_d$ and the gradient components in $v_k$:

$$\beta_d = (\beta_d, \gamma_d)^T \quad \text{and} \quad v_k = (v_k, w_k)^T.$$ 

A straightforward computation shows that

$$x_2' - x_1' = (\beta_d - \beta_d)\Delta + \alpha_d (w_2 - w_1) + (\alpha_d - \alpha_d) w_2$$

$$= (\gamma_d - \gamma_d)\Delta + \alpha_d (v_2 - v_1) + (\alpha_d - \alpha_d) v_1,$$

where $\Delta = y_2 - y_1$. We distinguish three different cases.

(i) $y_2 > x_2 > x_1 > y_1$: 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 (4.29) are zero, and $x_2' - x_1' = \alpha\Delta (w_2 - w_1)$.

(ii) $x_2 > x_1 > y_2$ or $x_2 > y_2 > x_1 > y_1$: observe that in both cases $d_2 > d_1$, $w_2 > 0$, and $w_2 - w_1 > 0$. If $\alpha_d > 0$ and $\beta_d, \gamma_d$ are non-increasing, then $(\beta_d - \beta_d)\Delta > 0$, $\alpha_d (w_2 - w_1) > 0$, and $(\alpha_d - \alpha_d) w_2 > 0$. Hence, we get from (4.29) that $x_2' - x_1' > 0$. If $\alpha_d < 0$ and $\beta_d, \gamma_d$ are non-decreasing, then all terms in (4.29) are negative, and we get $x_2' - x_1' < 0$.

(iii) $y_2 > x_2 > y_1 > x_1$: in this situation we have $v_2, w_2 > 0$ and $v_1, w_1 < 0$. We distinguish between the case where $d_2 \geq d_1$ and the case where $d_2 < d_1$. If $d_2 > d_1$, we can use the same argument as in case (ii). If $d_2 < d_1$, we use the identity in (4.30). If $\alpha_d > 0$, and $\beta_d, \gamma_d$ are non-increasing, all terms in (4.30) are positive and we get $x_2' - x_1' > 0$. On the other hand, if $\alpha_d < 0$, and $\beta_d, \gamma_d$ are non-decreasing, all terms in (4.30) are negative, and we get $x_2' - x_1' < 0$. \(\Box\)

We point out that $\alpha_d > 0$ is the case which seems most useful in practice. The corresponding scheme decreases the influence of the neighbor samples $y_1$ and $y_2$ when the gradient is large. This corresponds to the intuitive idea that sharp transitions (e.g., edges in an image) should not be smoothed to the same extent as regions which are more homogeneous.

So far, we have only derived conditions which guarantee that perfect reconstruction is possible, but we have not yet given the corresponding reconstruction 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 4.4.3, that is, we assume that $\alpha_d > 0$ for all $d \geq 0$, and that $\beta_{d,1}, \beta_{d,2}$ are non-increasing with respect to $d$.

**Lemma 4.4.4.** Assume that $y_2 \geq y_1$ and let $\Delta = y_2 - y_1$ and $d = |x - y_1| + |x - y_2|$. The following relations hold:

$$x < y_1 \quad \iff \quad x' < y_1 + \beta_{d,2}\Delta$$

$$y_1 \leq x \leq y_2 \quad \iff \quad y_1 + \beta_{d,2}\Delta \leq x' \leq y_2 - \beta_{d,1}\Delta$$

$$x > y_2 \quad \iff \quad x' > y_2 - \beta_{d,1}\Delta.$$

**Proof.** Since the three cases cover the entire real axis, it suffices to prove the relations in one direction. Here we will prove the implications $\iff$. Under the given assumptions we have $\Delta \leq d$. We can easily establish the following identities:

$$x' = x - \beta_{d,2}v_2 - \beta_{d,1}v_1$$

$$= y_2 + \alpha_d v_2 - \beta_{d,1}\Delta$$

$$= y_1 + \alpha_d v_1 + \beta_{d,2}\Delta.$$
From (4.32) we get immediately that if and only if $\alpha_d v_1 < 0$, i.e., $v_1 = x - y_1 < 0$, then $x' < y_1 + \beta_{d,2}\Delta$. This proves the first relation. Similarly, (4.31) yields that $x' > y_2 - \beta_{d,1}\Delta$ if and only if $\alpha_d v_2 > 0$, that is, $v_2 = x - y_2 > 0$. This accounts for the third relation. As for the second one, when $y_1 \leq x \leq y_2$, we have that $d = \Delta$, $v_1 \geq 0$, $v_2 \leq 0$, and (4.31)-(4.32) yield that $y_1 + \beta_{d,2}\Delta \leq x' \leq y_2 - \beta_{d,1}\Delta$.

Similar results can be obtained for case (b) of Proposition 4.4.3 as well as for the case that $y_2 < y_1$.

The previous lemma is essential in the construction of an algorithm which performs the inversion step. Note that we do not know the explicit values of $\alpha_d$, $\beta_{d,1}$, $\beta_{d,2}$, but we do know how to express them as a function of $d$. Thus, in order to reconstruct $x$, we first need to recover $d$, and then compute the filter coefficients, after which we can invert (4.26). Let us first restrict ourselves to the case $y_1 \leq y_2$. Observe that

$$x \in [y_1, y_2] \iff x' \in [y_1 + \beta_{d,2}\Delta, y_2 - \beta_{d,1}\Delta].$$

Thus, if $x' \in [y_1 + \beta_{d,2}\Delta, y_2 - \beta_{d,1}\Delta]$, then

$$d = \Delta = y_2 - y_1,$$

and reconstruction becomes straightforward. If, however, $x' \not\in [y_1 + \beta_{d,2}\Delta, y_2 - \beta_{d,1}\Delta]$, then

$$d = |y_1 + y_2 - 2x| = |y_1 + y_2 - \frac{2}{\alpha_d}(x' - \beta_{d,1}y_1 - \beta_{d,2}y_2)|.$$

This can be rewritten as

$$\alpha_d d = |y_1 + y_2 - 2x' + (\beta_{d,1} - \beta_{d,2})(y_1 - y_2)|.$$

Assuming that this latter equation has a unique nonnegative solution $d$, reconstruction is straightforward. The other cases (Proposition 4.4.3(b) and/or $y_2 < y_1$) can be treated similarly, and we arrive at the following algorithm.

**Algorithm**

1. Compute $\Delta = |y_2 - y_1|$.
2. Compute coefficients $\alpha_{\Delta}, \beta_{\Delta,1}, \beta_{\Delta,2}$.
3. Compute the lower and upper limits, $Y$ and $Z$, as

$$Y = \min\{y_1 + \beta_{d,2}(y_2 - y_1), y_2 - \beta_{d,1}(y_2 - y_1)\}$$
$$Z = \max\{y_1 + \beta_{d,2}(y_2 - y_1), y_2 - \beta_{d,1}(y_2 - y_1)\}.$$

(Note: the ‘min’ and ‘max’ are needed to cover both cases $y_1 \leq y_2$ and $y_2 < y_1$.)
4. If \( x' \in [Y, Z] \) (which implies \( d = \Delta \)) put

\[
\beta_2 = \beta_{\Delta,2} \quad \text{and} \quad \beta_1 = \beta_{\Delta,1};
\] otherwise

\[
(4a) \ \text{compute} \ d \ \text{by solving} \quad \alpha_d \ d = |y_1 + y_2 - 2x' + (\beta_{d,1} - \beta_{d,2})(y_1 - y_2)|;
\]

\[
(4b) \ \text{put} \quad \beta_2 = \beta_{d,2} \quad \text{and} \quad \beta_1 = \beta_{d,1}.
\]

5. Compute \( x \) from

\[
x = \frac{x' - \beta_1 y_1 - \beta_2 y_2}{1 - \beta_1 - \beta_2}.
\]

**Example 4.4.5.** Consider the case

\[
\beta_{d,1} = \beta_{d,2} = \frac{\beta_0}{\sigma d + 1},
\] (4.33)

where \( 0 \leq \beta_0 < \frac{1}{2} \) and \( \sigma > 0 \). It follows immediately that the conditions in Proposition 4.4.3(a) are satisfied. Steps (1) to (3) of the above algorithm are straightforward, and they yield the coefficients

\[
\beta_{\Delta,1} = \beta_{\Delta,2} = \frac{\beta_0}{\sigma \Delta + 1} = \frac{\beta_0}{\sigma |y_2 - y_1| + 1}.
\]

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 \( \beta_1 = \beta_2 = \beta_{\Delta,1} \). Now we can retrieve \( x \) following step (5). Otherwise, we must solve the equation given in step (4a) with \( \alpha_d = 1 - 2 \beta_{d,1} \):

\[
(1 - 2 \beta_{d,1})d = |y_1 + y_2 - 2x'|.
\]

Expressing \( \beta_{d,1} \) as a function of \( d \), and denoting \( r = |y_1 + y_2 - 2x'| \), we arrive at the quadratic equation \( \sigma d^2 + (1 - 2 \beta_0 - r \sigma)d - r = 0 \). This equation has a unique positive solution:

\[
d = \frac{-(1 - 2 \beta_0 - r \sigma) \pm \sqrt{(1 - 2 \beta_0 - r \sigma)^2 + 4 r \sigma}}{2 \sigma}.
\]

From this \( d \) we can compute the filter coefficients \( \beta_1 = \beta_2 := \beta_{d,1} \), and retrieve \( x \) using step (5).

As a particular instance of the continuous case presented in this section, we derive a binary scheme such as the one studied in Section 4.3 for the \( l^1 \)-norm. Consider the coefficients given by

\[
\beta_{d,j} = \begin{cases} 
\beta_{0,j} & \text{if } d \leq T \\
\beta_{1,j} & \text{if } d > T 
\end{cases} \quad \text{for } j = 1, 2. \quad (4.34)
\]

Proposition 4.4.3 yields the following result.
4.5. Other cases

**Corollary 4.4.6.** Assume an update lifting step as in (4.26) where the decision \( d \) is given by (4.28) and the coefficients \( \beta_{d,1}, \beta_{d,2} \) are given by (4.34). Perfect reconstruction is guaranteed in each of the following two cases:

(a) \( \alpha_0 > 0, \beta_{0,1} \geq \beta_{1,1}, \beta_{0,2} \geq \beta_{1,2} \).

(b) \( \alpha_1 < 0, \beta_{0,1} \leq \beta_{1,1}, \beta_{0,2} \leq \beta_{1,2} \).

Note, however, that these requirements are not enough if we wish to use the reconstruction algorithm described in Section 3.5 (page 56), since the threshold criterion does not hold, i.e.,

\[
|v_1| + |v_2| > T \Rightarrow |v'_1| + |v'_2| > T',
\]

\[
|v_1| + |v_2| \leq T \Rightarrow |v'_1| + |v'_2| \leq T'.
\]

4.4.2 Simulations

We consider the case where the coefficients are given by (4.33) in Example 4.4.5, with \( \sigma = 5 \). That is,

\[
\beta_{d,1} = \beta_{d,2} = \frac{1}{5d + 1} = \frac{1}{15d + 3}.
\]

Note that for small values of \( d \), the resulting low-pass filter approximates the average filter (\( \alpha = \beta_1 = \beta_2 = \frac{1}{3} \)), while for large values it behaves more like the identity filter (\( \alpha = 1, \beta_1 = \beta_2 = 0 \)).

Thus, again we can consider these two extreme cases to be the non-adaptive counterparts of the proposed scheme. We use the same input signal and prediction step as in Experiment 4.3.1 (corresponding with Fig. 4.14). The two bottom rows of Fig. 4.14 correspond with the two aforementioned extreme cases. The approximation and detail signals resulting from our scheme are depicted at the bottom of Fig. 4.15. The corresponding decision map, which in this case equals the \( l^1 \)-norm gradient, is shown at the middle left. On the right, the corresponding coefficients \( \beta_{d,1} = \beta_{d,2} \) have been depicted. These figures show clearly that sharp transitions are smoothed to a much lesser degree than more homogeneous parts of the signal.

4.5 Other cases

Up to this point we have considered linear update filters and two different decision maps based on the gradient vector \( v \in \mathbb{R}^N \). In particular, we have studied the cases where the output of the decision map is given by:

(a) \( d = [p(v) > T] \) where \( p \) is a given seminorm and \( T \) is a threshold;

(b) \( d = p(v) \) where \( p \) is the \( l^1 \)-norm and \( N = 2 \).

Nevertheless, we can also think of using nonlinear update filters and/or other decision maps while keeping the perfect reconstruction condition. In this section, we give some examples of
such adaptive schemes. In all cases, as it will be shown, the resulting updated sample $x'$ can be written as

$$x' = \alpha_d x + \sum_{j=1}^{N} \beta_{d,j} y_j$$

with $\kappa_d = \alpha_d + \sum_{j=1}^{N} \beta_{d,j} = 1$ for all $d \in D$.

### 4.5.1 Ranking-based updating

In this subsection, we give some examples of adaptive update lifting schemes using ranking operators such as the minimum and the median operators.
Using the minimum operator

A simple yet intuitive example of an adaptive update lifting step is to update the input sample $x$ with the neighbor $y_j$, $j \in \{1, \ldots, N\}$, which is closest in value to $x$. That is,

$$
x' = \alpha x + \beta y_s, \quad \text{with } s = \text{argmin}_j |x - y_j|.
$$

(4.35)

It may occur that more than one sample $y_j$ minimizes this distance. In this case, we consider, e.g., the one with lowest $j$.

Assume that

$$
\text{argmin}_j |x - y_j| = \text{argmin}_j |x' - y_j|
$$

with $x'$ given by (4.35). Then, it is obvious that we can recover $x$ by

$$
x = \frac{1}{\alpha}(x' - \beta y_s), \quad \text{with } s = \text{argmin}_j |x' - y_j|.
$$

(4.37)

We show that if $0 \leq \beta \leq 1$, then (4.36) is always satisfied and thus we can invert (4.35). First observe that

$$
|x' - y_j| = |(1 - \beta)x + \beta y_s - y_j| = |x - y_j - \beta(x - y_s)|.
$$

(4.38)

If $y_j = y_s$, then $x - y_j = x - y$, and

$$
|x' - y_j| = (1 - \beta)|x - y_s|.
$$

If $y_j \neq y_s$, there are two possibilities:

(i) $|x - y_j| = |x - y_s|$: thus $x - y_j = y - x$ and

$$
|x' - y_j| = (1 + \beta)|x - y_s|.
$$

(ii) $|x - y_j| > |x - y_s|$: from (4.38) it obvious that

$$
|x' - y_j| \geq |x - y_j| - \beta|x - y_s| > (1 - \beta)|x - y_s|.
$$

Therefore, if $s = \text{argmin}_j |x - y_j|$ and $\beta \in [0, 1)$, we have that

$$
|x' - y_j| \geq (1 - \beta)|x - y_s| \quad \text{for all } j \in \{1, \ldots, N\},
$$

where the equality holds only if $y_j = y_s$. Thus, condition (4.36) is satisfied.

The above result can be generalized using more than one closest neighbor to update $x$. As an example, we consider the case where $N = 2$ and the update step is given by

$$
x' = \begin{cases} 
\frac{1}{2}x + \frac{1}{2}y_1 & \text{if } |v_1| < |v_2| \\
\frac{1}{2}x + \frac{1}{4}(y_1 + y_2) & \text{if } |v_1| = |v_2| \\
\frac{1}{2}x + \frac{3}{2}y_2 & \text{if } |v_1| > |v_2|.
\end{cases}
$$
Chapter 4. Adaptive update lifting: specific cases

where we have used the gradient notation $v_j$ instead of $x - y_j$. This scheme has the following simple interpretation. The updated sample $x'$ is obtained by averaging the original sample $x$ with the neighbor which is closest in value. If both neighbors $y_1$, $y_2$ are equally close, we average $x$ with the average of both neighbors.

This is equivalent to consider the update step $x' = \alpha x + \beta_{d,1} y_1 + \beta_{d,2} y_2$ with the three-valued decision map

$$d = \text{sign}(|v_1| - |v_2|),$$

where $\text{sign}(0) = 0$, and coefficients $\beta_d$ given by

$$\beta_{-1} = \frac{1}{2} (1, 0)^T, \quad \beta_0 = \frac{1}{2} (1, 1)^T, \quad \beta_{+1} = \frac{1}{2} (0, 1)^T.$$ 

We demonstrate that we can recover $d$ at synthesis and therefore have perfect reconstruction.

If $d = -1$ at analysis (i.e., $|v_1| < |v_2|$), then $|v_2'| = \frac{1}{2} |v_1|$ and we derive that

$$|v_2'| = \frac{1}{2} (x + y_1) - y_2 = |v_2 - \frac{1}{2} v_1| \geq |v_2| - \frac{1}{2} |v_1| > \frac{1}{2} |v_1| = |v_1'|,$$

which results in $d = -1$ at synthesis. If $d = +1$ at analysis (i.e., $|v_1| > |v_2|$), a similar reasoning yields $|v_1'| > |v_2'|$, and thus $d = +1$ at synthesis.

If $d = 0$ (i.e., $|v_1| = |v_2|$), there are two possibilities:

1. $y_1 = y_2$: it is obvious that $|v_1'| = |v_2'|$, hence $d = 0$ at synthesis.
2. $v_1 = -v_2$: then $x' = x$, and therefore $d = 0$ at synthesis.

**Experiment 4.5.1 (Updating with closest neighbors for 1D, $N = 2$ - Fig. 4.16)**

We illustrate the performance of this adaptive system in the 1D case where sample $x$ corresponds to $x(n) = x_0(2n)$ and samples $y_1$, $y_2$ to $y(n-1) = x_0(2n-1)$ and $y(n) = x_0(2n+1)$, respectively. As in previous 1D simulations, we assume a fixed prediction step of the form given in (4.12). The input signal is shown at the top left of Fig. 4.16. The middle row shows the decomposition results from the adaptive scheme, while the last row depicts the results from the non-adaptive scheme with fixed $d = 0$.

**Using the median operator**

Now, consider the adaptive update lifting step:

$$x' = \alpha x + \beta z, \quad \text{with } z = \text{median}\{x, y_1, y_2, \ldots, y_N\}$$

and $N$ an even number. We can show that if $\beta \in [0, 1)$ we can recover $x$ by

$$x = \frac{1}{\alpha} (x' - \beta z), \quad \text{with } z = \text{median}\{x', y_1, y_2, \ldots, y_N\}.$$ 

The proof is straightforward if we observe that

$$\text{median}\{x, y_1, y_2, \ldots, y_N\} = \text{median}\{x', y_1, y_2, \ldots, y_N\}.$$
where $N$ is even and $x'$ is given by (4.39).

Experiment 4.5.2 (Threshold criterion and median-based update for 1D, $N = 2$ - Fig. 4.17)
We consider the adaptive update lifting step

$$x' = a_d x + \beta_d \text{median}(x, y_1, y_2),$$

where $d = [p(v) > T], v \in \mathbb{R}^2$ and $p(v) = \min(|v_1|, |v_2|)$. We choose $\beta_1 = (0, 0)^T$ and $\beta_0 = \frac{1}{2}(1, 1)^T$. Then we can write (4.40) as

$$x' = \begin{cases} 
\frac{1}{2} x + \frac{1}{2} \text{median}(x, y_1, y_2) & \text{if } d = 0 \\
\frac{1}{2} x & \text{if } d = 1.
\end{cases}$$

It is easy to check that we can recover the decision at synthesis.
The simulation results are shown in Fig. 4.17 where two levels of decompositions and a threshold of $T = 0.5$ have been used. As in the previous experiment, sample $x$ corresponds to $x(n) = x_0(2n)$ and samples $y_1, y_2$ to $y(n-1) = x_0(2n-1)$ and $y(n) = x_0(2n+1)$, respectively, and the fixed prediction step is of the form given in (4.12). The first row of Fig. 4.17 shows the input signal (left) and the decision map (right) at level 1. The second row shows the approximation (left) and detail (right) signals at level 2. The vertical dotted lines in the approximation subfigure represents the decision map at level 2. For comparison, the decomposition obtained for both non-adaptive cases corresponding with fixed $d = 0$ (median-based filter) and $d = 1$ (identity filter) are shown in the third and fourth rows, respectively. One can observe that in the adaptive scheme, the approximation signal is smoothed by the median-based filter except in those locations where there are sharp transitions (hence $d = 1$), in which case no filtering is performed. This could be useful for detecting sharp transitions while removing noise.

4.5.2 Switching between horizontal and vertical filters

In this subsection we consider 2D signals. We build a decision map that uses two seminorms, one governing the horizontal gradient and one for the vertical gradient. Taking $N = 4$, we choose the samples $y_j$ in such a way that they correspond with the four horizontal and vertical neighbors of sample $x$. We use the labeling shown in Fig. 4.18. Thus, only the horizontal and vertical neighbors $y_1, y_2, y_3, y_4$ are used to update $x$. Note that this may correspond either to a quincunx or to a square decomposition scheme.

We define

$$p_h(v) = |v_1 + v_3| \quad \text{and} \quad p_v(v) = |v_2 + v_4|,$$

corresponding, respectively, with a horizontal and a vertical derivative filter of second order. We choose a binary decision map whose output $d$ is given by

$$d = [p_h(v) \leq p_v(v)] . \quad (4.41)$$

The filter coefficients $\beta_d$ are chosen as follows:

$$\beta_d = (\beta_d, \gamma_d, \beta_d, \gamma_d)^T \quad \text{for } d = 0, 1 . \quad (4.42)$$

For example, if $d = 1$, then the update of sample $x$ is

$$x' = \alpha_1 x + \beta_1 (y_1 + y_3) + \gamma_1 (y_2 + y_4).$$

This case has the following geometric interpretation. If $p_h(v) \leq p_v(v)$, and hence $d = 1$, then the vertical derivative $2x - y_2 - y_4$ dominates (in absolute value) the horizontal derivative $2x - y_1 - y_3$, and in this case we choose the filter in such a way that it causes a stronger smoothing in the horizontal than the vertical direction, i.e., $\gamma_1 < \beta_1$.

If we can choose the coefficients $\beta_0, \gamma_0, \beta_1, \gamma_1$ in such a way that

$$[p_h(v) \leq p_v(v)] = [p_h(v') \leq p_v(v')] ,$$

then we can recover the original decision in (4.41) from the gradient vector at synthesis $v'$, and hence perfect reconstruction is possible in this case.
4.5. Other cases

Figure 4.17: Decompositions (at level 2) corresponding with Experiment 4.5.2. Top: original signal (left) and decision map at level 1 (right). Second row: approximation (left) and detail (right) signals in the adaptive case. Third and bottom rows: approximation (left) and detail (right) signals in the non-adaptive cases with $d = 0$ (median-based filter) and $d = 1$ (identity filter), respectively.
Chapter 4. Adaptive update lifting: specific cases

Proposition 4.5.1. To have perfect reconstruction it is sufficient that

\[ 0 < \beta_0 < \frac{1}{4} \leq \gamma_0 < \frac{1}{2} \quad \text{and} \quad \beta_0 + \gamma_0 < \frac{1}{2}, \]

\[ 0 < \gamma_1 < \frac{1}{4} \leq \beta_1 < \frac{1}{2} \quad \text{and} \quad \beta_1 + \gamma_1 < \frac{1}{2}, \]

and in this case the decision \( d \) can be recovered at synthesis from
\[ d = \left[ p_v(v') \leq p_v(v') \right]. \]

Proof. We introduce the following notation for the horizontal and vertical components of the gradient:

\[ H = v_1 + v_3 \quad \text{and} \quad V = v_2 + v_4, \]

and the same for \( H' \) and \( V' \). A straightforward computation shows that

\[ H' = (1 - 2\beta_1)H - 2\gamma_1 V \]

\[ V' = -2\beta_1 V - (1 - 2\gamma_1)H. \]

We will prove that if the decision map returns \( d = 1 \), i.e., \( |H| \leq |V| \), then it also follows that \( |H'| \leq |V'| \). The proof for the case where the decision map returns \( d = 0 \) is analogous. We distinguish four different cases.

(i) \( 0 \leq H < V \): then

\[ 0 < V' = 2\beta_1(V - H) + (1 - 2\gamma_1 - 2\beta_1)V \]

\[ H' = -(1 - 2\beta_1)(V - H) + (1 - 2\gamma_1 - 2\beta_1)V. \]

Since \( (1 - 2\gamma_1 - 2\beta_1)V \) is positive and

\[ 0 < (1 - 2\beta_1)(V - H) \leq 2\beta_1(V - H), \]

it follows that \( |H'| \leq |V'| \).
(ii) $H \geq 0$ and $V < 0$: thus $H'$ in (4.43) comprises two positive terms, whereas $V'$ in (4.44) comprises two negative terms. Since $1 - 2\beta_1 < 2\gamma_1$ and $2\gamma_1 < 1 - 2\gamma_1$, it is obvious that $|H'| \leq |V'|$.

(iii) $H < 0$ and $V \geq 0$: now $H'$ in (4.43) comprises two negative terms and $V'$ in (4.44) comprises two positive terms. The same reasoning as in (ii) yields that $|H'| \leq |V'|$.

(iv) $H < 0$ and $V < 0$: from $|H| \leq |V|$ we conclude that $-H \leq -V$, hence $H - V \geq 0$. Now

\[
H' = (1 - 2\beta_1)(H - V) + (1 - 2\beta_1 - 2\gamma_1)V \\
V' = -2\beta_1(H - V) + (1 - 2\beta_1 - 2\gamma_1)V .
\]

Using a similar argument as in (i) we conclude again that $|H'| \leq |V'|$. \qed

**Experiment 4.5.3 (Switching between horizontal and vertical filters - Fig. 4.19)**

We choose the filter coefficients like in (4.42) with $\beta_0 = \gamma_1 = 0$ and $\beta_1 = \gamma_0 = 1/4$. Obviously the conditions in Proposition 4.5.1 are satisfied. We consider a 2D square sampling scheme such as depicted in Fig. 4.6. After the update step, we perform the prediction as in (4.14)-(4.15).

We apply this scheme to the original image depicted at the top left of Fig. 4.19. The decision map is shown at the top right; the approximation and horizontal detail images are shown in the second row. The diagonal detail is displayed in the bottom row, on the left. We compare this scheme with the non-adaptive scheme where we perform an isotropic filtering (in the vertical and horizontal directions), i.e., $\beta = \gamma = 1/8$. The corresponding approximation and horizontal images are displayed in the third row of Fig. 4.19, and the diagonal detail image on the right of the bottom row. We can easily see that the approximation image obtained in the adaptive case preserves the edges in contrast with the one obtained in the non-adaptive scheme. Consequently, the detail images obtained in the adaptive case 'capture' the edges in a more compact way that in the non-adaptive case.
Figure 4.19: Decompositions (at level 1) corresponding with Experiment 4.5.3. Top: input image (left) and decision map (right). Second row: approximation (left) and horizontal detail (right) images in the adaptive case. Third row: approximation (left) and horizontal detail (right) images in the non-adaptive case with $d = 0$. Bottom: diagonal detail images in the adaptive (left) and non-adaptive (right) cases.