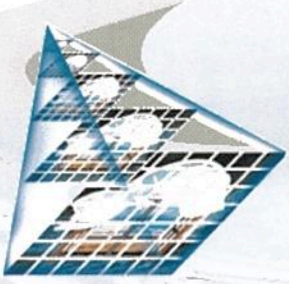
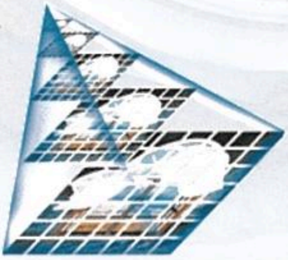
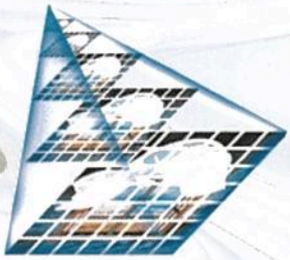
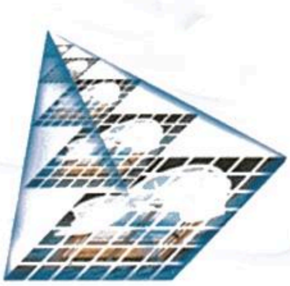






ADAPTIVE WAVELETS AND THEIR APPLICATIONS TO IMAGE FUSION AND COMPRESSION

GEMMA PIELLA



Adaptive Wavelets
and their Applications to
Image Fusion and Compression

Gema Piella Fenoy

Adaptive Wavelets
and their Applications to
Image Fusion and Compression

Cover design by T. Baanders.
Printed and bound by Ponsen & Looijen bv.

Adaptive Wavelets and their Applications to Image Fusion and Compression

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. mr. P.F. van der Heijden
ten overstaan van een door het college voor promoties ingestelde
commissie, in het openbaar te verdedigen in de Aula der Universiteit
op donderdag 30 oktober 2003, te 10.00 uur

door

Gema Piella Fenoy

geboren te Barcelona, Spanje

Promotor:
prof. dr. P.W. Hemker

Co-promotor:
dr. ir. H.J.A.M. Heijmans

Faculteit der Natuurwetenschappen, Wiskunde en Informatica
Universiteit van Amsterdam

The research reported in this thesis has been carried out at the Centre for Mathematics and Computer Science (CWI), Amsterdam.

The work has been financially supported by the Dutch Technology Foundation STW.

to the memory of my father Jaume Piella

Contents

Chapter 1

Introduction to multiresolution signal processing

There can be many different ways to represent a signal and the usefulness of a given representation depends on the subsequent interpretation or manipulation of its content. If the signal needs to be compressed, one is interested in representations that are sparse: obviously the description $(-1)^n + n$, $n = 1, 2, \ldots, 1000$, needs less bits than $0, 3, 2, \ldots, 1001$. For denoising applications, one is interested in representations that separate the robust global characteristics of the signal from the random-like local fluctuations. The challenge is to find a transform that is intrinsically well adapted to represent a signal for a given application.

In the last decades, new ways of representing a signal have been developed; among those, multiresolution representations occupy a prominent place and have proved to be a powerful tool both from a theoretical and a practical points of view.

This chapter provides a short overview to multiresolution representations and their application to signal processing. It serves both as an introduction and as a motivation for further discussions in the next chapters. Special attention is paid to the pyramid and the wavelet representations, as well as to the application of multiresolution representations in image fusion. In the final section, we give an outline of this thesis.

1.1 The need for multiresolution representations

1.1.1 Classical approaches

There are several ways to transform one representation of a given signal into another one. The most classical example is the Fourier transform [11, 58], where a signal is decomposed into sinusoidal waves. Such a decomposition gives the intensity of the fluctuations (frequencies) in the signal which is often of great importance. However, due to the infinite extent of the sinusoidal functions, any local signal characteristics (i.e., an abrupt change in the signal) are spread over the entire representation, thus making them 'invisible'. This is a serious drawback since singularities and irregular structures often carry the most important information in signals. For instance, in images, discontinuities in the intensity may provide the location of the object contours which are particularly meaningful for recognition purposes. For many other types of signals such as electro-cardiograms or radar signals, the interesting information is given by transients such as local extrema. Furthermore, such singularities usually occur with different location and localization (i.e., range, scale) in time and frequency. Consequently, transform methods that represent the signal at multiple scales are better suited for extracting information than methods that represent the signal at a single scale.

Over the last century, scientists in different fields struggled to overcome limitations of the Fourier transform and to build representations of signals that are able to adapt themselves to the nature of the signal. On the one hand, to 'pick up' the transients without giving up the frequency information, the signal should be decomposed over functions which are well localized in time (or space) and frequency. This leads to so-called *time-frequency representations*. On the other hand, since signal structure depends on the scale at which the signal is being perceived, it should be analyzed at different scales or levels of resolution. This results in so-called *multiresolution representations* which, besides a time parameter, also contain a scale parameter.

1.1.2 Time-frequency trade-off

As observed in the previous subsection, the form in which a signal is represented is important in any signal processing task. Different representations emphasize different aspects of a signal and therefore, one should look for representations that make relevant information easily accessible. This depends on the signal and on the subsequent processing tasks. In general, one would like to represent a signal with a small number of well-defined basis functions. For example, a pure sinusoidal wave of frequency w_0 appears as a peak at frequency w_0 in the frequency domain. Thus, in this case the Fourier transform represents the signal in a much more revealing and concentrated form than a traditional time representation.

The motivation behind time-frequency representations is that by jointly representing the signal in both domains, one can reveal the behavior of the signal in both the time and the frequency domain and develop signal processing tools that exploit this information. Before introducing some of these representations, let us examine the extreme cases of a pure time-domain and a pure frequency-domain description of a signal.

A one-dimensional signal in time can be considered as a representation in terms of Dirac functions. These functions are perfectly localized in time but have no frequency localization (see Fig. 1.1(a)). Thus, such a time representation gives the exact value of the signal at each location, but it does not provide any information about the underlying frequency content. Alternatively, the signal can be represented in terms of its frequency components by means of the Fourier transform. In this case, the functions used in the representation are pure harmonic waves, which are perfectly local in frequency but have no time localization because of their infinite extent in time (see Fig. 1.1(b)). Thus, a frequency representation reveals the frequency components of a signal but hides its local temporal behavior. Yet, both representations are mathematically equivalent in the sense that all information is present in each representation (either in time or in frequency). The difference consists hereof that they give access to different features.

Often, one wants to have access to both time and frequency information simultaneously.

(a) (b)

Figure 1.1: *Localization in time and frequency: (a) Dirac bases; (b) Fourier bases.*

This can be accomplished by decomposing the signal in terms of functions that are both localized in time and frequency. In this way, a one-dimensional signal is mapped onto a two-dimensional function in the time-frequency plane (t, w).

There are various ways to define the *localization* of a function, but they are all related to the spread of the function in time and frequency. Given a finite energy function f with Fourier transform F, one can think of reducing its time spread by scaling with a factor $0 < a < 1$, i.e., $\tilde{f}(t) = f(t/a)$. For the Fourier transform $\tilde{F}$, this amounts to a dilation, that is: $\tilde{F}(w) = aF(aw)$. Thus, we have lost in frequency localization, what we have gained in time. Conversely, if we reduce the frequency spread by a factor $a > 1$, we increase the time spread by a. This suggests that there is a trade-off between time and frequency localization. Indeed, both localizations are subjected to the *uncertainty principle* [59], which formalizes mathematically the time-frequency trade-off observed before. More precisely, consider a signal $f \in L^2(\mathbb{R})$ (i.e., finite energy) with Fourier transform F, and define

$$\tau = \frac{1}{\|f\|^2} \int_{-\infty}^{\infty} t|f(t)|^2 dt\,, \qquad u = \frac{1}{2\pi\|F\|^2} \int_{-\infty}^{\infty} w|F(w)|^2 dw\,,$$

and

$$\sigma_t^2 = \frac{1}{\|f\|^2} \int_{-\infty}^{\infty} (t-\tau)^2|f(t)|^2 dt\,, \qquad \sigma_w^2 = \frac{1}{2\pi\|F\|^2} \int_{-\infty}^{\infty} (w-u)^2|F(w)|^2 dw\,.$$

In these expressions, $\|\cdot\|$ denotes the L^2-norm. The Heisenberg's uncertainty principle states that

$$\sigma_t^2 \sigma_w^2 \geq \frac{1}{4}\,,$$

where the equality holds only for Gaussian signals.

The localization of f in the time-frequency plane (t, w) can be represented by a tiling rectangle centered at (τ, u) whose width along time is σ_t and whose width along frequency is σ_w. This is illustrated in Fig. 1.2. Such a tile defines the *joint resolution*[1] in time and frequency of f.

The uncertainty principle limits the joint resolution in time and frequency of any linear signal representation: the frequency content at a certain time can only be known with finite accuracy.

---

[1] Here resolution refers to the degree of precision to which a quantity (e.g., time or frequency) can be measured or determined.

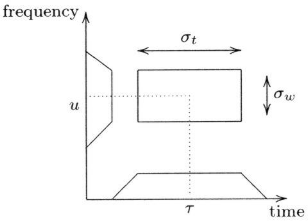


Figure 1.2: *Tile in the time-frequency plane as an approximation of the time-frequency localization of a signal. The tile contains most of the energy of the time-domain and frequency-domain representation of the signal.*

One of the first time-frequency representations is the windowed Fourier transform or short-time Fourier transform. This transform replaces the Fourier transform's sinusoidal wave by the product of a sinusoid and a smooth window localized in time. The resolution of the windowed Fourier transform depends on the spread of the window in time and frequency. However, once a window has been chosen, the time-frequency resolution is fixed over the entire time-frequency plane since the same window is used for all frequencies; see Fig. 1.3(a). The basic idea behind multiresolution signal representations is to vary the shape of the windows in the time-frequency tiling such as in Fig. 1.3(b)-(c).

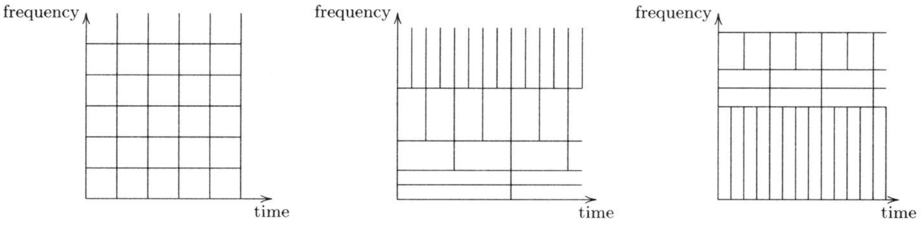
The most famous example in this respect is the wavelet transform. It replaces the Fourier's transform sinusoidal waves by a family generated by translations and dilations of a basis signal called wavelet[2] function. This results in a family of functions with varying time-frequency localization. As a result, the wavelet transform analyzes a signal at different frequencies with different time resolutions. High frequencies are analyzed with short temporal windows, while low frequencies are analyzed with longer windows; see Fig. 1.3(b). Thus, it offers a good time resolution at high frequencies, and a good frequency resolution at low frequencies. This time-frequency trade-off makes sense especially when the signal at hand has high-frequency components of short durations and low-frequency components of long durations. Indeed, many real-world signals do exhibit such type of behavior.

A still greater diversity in time-frequency tilings (see e.g. Fig. 1.3(c)) can be obtained by introducing the notion of a wavelet packet; we refer to Section 2.5 for more details.

In wavelet theory, one speaks about time-scale representations rather than time-frequency representations, scale being in a way the inverse of frequency, because the term frequency is reserved for the Fourier domain. In Section 1.4, we describe the wavelet transform in more detail.

---

[2] The term wavelet means 'small wave'. 'Small' refers to its finite length (or energy) in time, i.e., good localization in time. 'Wave' refers to its oscillatory behavior in time, i.e., good localization in frequency.



(a) (b) (c)

Figure 1.3: *Time-frequency tilings: (a) windowed Fourier basis; (b) wavelet basis; (c) wavelet packet basis.*

1.1.3 Multiresolution: what is it good for?

Multiresolution representations make it possible to analyze signals at multiple scales. One way of thinking about multiresolution processing is to consider how our eyes look at the world. One can observe a scene such as a forest at different scales. As we get closer, we can distinguish the individual trees, then the branches and finally the leaves. As we zoom in at smaller and smaller scales, we can see details that we did not see before, but at the same time the view field is reduced.

Signals often contain physically relevant features at many different scales or resolutions. Thus, for a comprehensive understanding of a signal, one has to analyze it over a broad range of resolutions. Sometimes signals behave in a similar way across different scales (i.e., scale-invariance). For signals that are not scale-invariant, one does not know in general at which scales the signal contains the more relevant information. Therefore all scales are of equal importance a priori. This suggests that for a proper interpretation of the signal information, the signal should be analyzed at different scales in a uniform way, i.e., the size of the operators used to analyze the signal must be adapted to the size of the structures contained within the signal. This size defines the inner scale or *resolution*, which is the smallest detail that can be distinguished.

A representation of a signal which can be subdivided into parts, each of which corresponds with a given resolution, is called a *multiresolution representation*. The structures at high scales in a multiresolution (MR) representation correspond to low resolutions (coarse details) of corresponding structures at lower scales (finer details). Thus, MR representations, often referred to as MR decompositions, allow us to see the 'forest and the trees', so to speak.

There are many reasons for taking recourse to MR techniques. First, as we already pointed out, most signals exhibit relevant features at many different resolutions. For example, images contain objects of different sizes. Moreover, there are similarities between MR processing and the way the human auditory and visual system work [102,110]. Secondly, it may be the case that data is available at multiple resolutions. For instance, sensors may provide signals at different spatial and frequency resolutions. Third, there may be a need for output at different multiple resolutions. This is the case in multimedia applications where a signal may be shared by several target output devices with different resolutions. There are other compelling reasons to consider MR techniques. In particular, the resulting algorithms may offer computational advantages. This can be seen in a variety of methods for the solution of large systems of equations. Another important feature of MR approaches is that they allow for interaction between different levels, which is very useful in numerous applications such as image segmentation.

For a brief discussion on different MR techniques and applications see Sections 1.2.2 and 1.2.3.

1.2 Multiresolution approaches

MR methods span a very broad array of concepts and approaches, and in this section we examine some of them.

1.2.1 Continuity versus discreteness, redundancy versus non-redundancy

In Section 1.1.2, we implicitly assumed continuous-time signals, but similar observations can be made for discrete signals.

The theory of time-frequency representations in $L^2(\mathbb{R})$ (in particular wavelets) has been developed in the general context of frames[3] [43]. One of the important results was the 'discovery' of the relationship between continuous wavelet representations in $L^2(\mathbb{R})$ and their discrete-time analog, which had been developed independently in the framework of filter banks and subband coding [38, 139, 162]. For example, iteration of discrete-time filter banks converges, under certain conditions, to a continuous-time wavelet basis [44], enabling efficient computation of corresponding continuous-time wavelet coefficients [98]. Likewise, oversampled filter banks can be studied using the theory of frames, and filters banks which do not involve subsampling implement transforms similar to continuous-time transforms [40].

These observations reveal a close relation between discrete and continuous-time wavelet transforms. Frame theory [54] provides a general framework which links the two extreme situations of 'everything continuous and redundant' and 'everything discrete and non-redundant'. Reconstruction of the original signal is possible under some restrictions. If the redundancy is large, then only mild restrictions are put on the functions used in the representation. Conversely, if the redundancy is small, then much more constraints are imposed on such functions.

In Section 1.4, we examine the links between discrete and continuous, and between redundant and non-redundant representations for the wavelet case.

---

[3] Stated somewhat imprecisely, a set of vectors in a vector space is called a frame if it is complete but not necessarily independent. Because of the latter, frame representations are redundant in general.

1.2.2 Examples of multiresolution representations and methods

In this thesis, we will mainly focus on pyramid and wavelet representations. There are, however, many other MR techniques. They have the 'multiresolution paradigm' in common, but apart from that they differ in many respects, both in theory and in practice. In this subsection we list some MR approaches as an illustration of the wide variety of methods.

Pyramids

Pyramids have been recognized early as an interesting tool for computer vision and image coding [1, 19]. A classical pyramid scheme consists of three steps: (*i*) deriving a coarse approximation of an input image, (*ii*) predicting this image based on the coarse version, and (*iii*) taking their difference as the prediction error. This defines the analysis part. At synthesis, the prediction error is added back to the prediction from the coarse version, guaranteeing perfect reconstruction. Iteration of the analysis part over the coarse approximation yields a pyramid representation of the original image as an approximation image at the lowest resolution and a set of detail images at successive higher resolutions. A special case of pyramid representation is studied in more detail in Section 1.3; see also Section 2.3 for a general framework.

Wavelets

As mentioned in Section 1.1.2, wavelets are functions that are well localized in time and frequency and that can be used to decompose a signal into different frequency bands with different time resolutions. This leads to the wavelet transform. Of particular interest is the discrete wavelet transform, which applies a two-channel filter bank (with downsampling) iteratively to the low-pass band (initially the original signal). The wavelet representation consists then of the low-pass band at the lowest resolution and the high-pass bands obtained at each step. This transform is invertible and non-redundant. As such, the corresponding decomposition differs from various other MR decompositions such as pyramids, which are redundant, and scale-spaces (see below), which are non-invertible in general. Both aforementioned properties, i.e., invertibility and non-redundancy, turn the discrete wavelet transform into a highly efficient and applicable representation for a broad range of signal and image processing tasks such as denoising and, particularly, compression. Wavelet decompositions will be studied thoroughly throughout this thesis.

Quadtrees

Quadtrees [80] divide an image into successively smaller quadrants. If the quadrant does not satisfy some 'homogeneity' criterion (e.g., all pixels have similar value), then it is subdivided, and so on, until all quadrants are 'homogeneous'. This process is easily captured by a tree structure. The entire image is represented by a root node, while the subsequent quadrants are child nodes, in a predetermined order. The leaf nodes correspond to those quadrants for which no further division was necessary. Quadtrees provide a coarse-to-fine approach which can significantly speed up, e.g., recognition and segmentation algorithms.

Multigrid methods

Multigrid methods [64] form an important tool in computational mathematics. Their key principle is to use coarser (and hence computational simpler) versions of a problem to guide the solutions of finer versions, with such finer versions used in turn to correct for errors in the coarser versions. Thus, multigrids move back and forth between scales as they solve a problem on the fine scale. They provide an effective and very flexible technique to model an unknown function (e.g., the solution to a partial differential equation).

Scale-space

Scale-space representations form a special type of MR representation that comprise a continuous scale parameter and preserve the same spatial sampling at all scales. The introduction of scale-space theory by Witkin [170] and Koenderink [81] was a breakthrough in image understanding. These authors were among the first to realize that the notion of 'scale' was coupled to a variety of basic concepts, such as the smallest entity of an image containing discernible information (the inner scale or resolution), and the regularized calculation of an image derivative.

Fractal imaging

Fractals are signals containing self-similarities, that is, they can be subdivided in parts, each of which is (at least approximately) a scaled copy of another part of the signal. Fractal signal analysis is based on the observation that in many real-world signals and processes, there occur patterns that repeat themselves at different scales. Fractal models try to identify and represent self-similarity relationships within images through the use of spatial transformations of the signal associated with shrinking, translation, symmetries and scaling. These transformations are particularly suited to be interpreted in an MR fashion. In the past, a lot of research effort was put in the field of fractal coding of images [77]. The most important concept here was the *iterated function system* (IFS) that was popularized by Barnsley [5].

In the literature one finds various other MR methods. For example, many of the existing morphological techniques [140], such as granulometries, skeletons and alternating sequential filters are essentially MR techniques.

1.2.3 Applications of multiresolution representations

We describe a few typical applications of MR signal processing.

Compression

MR decompositions allow for sparse signal representation capturing the essence of the signal with only a small set of significant coefficients. This is due to the fact that most signals have correlation both in space and frequency. Moreover, some compression methods take advantage of the limitations of our perceptual system to achieve high compression ratios without noticeable degradation. For example, the human visual system tends to be more sensitive to errors in low-frequency image components than in high-frequency ones. Therefore detail coefficients at higher levels can be omitted without causing serious degradation of the approximated image quality.

Pattern recognition

Suppose we need to locate a large complex pattern within an image. Rather than attempting to convolve the whole pattern with the image, one may perform an approximate search by convolving a reduced-resolution pattern with a reduced-resolution version of the image. Thus one can roughly locate possible occurrences of the target pattern with a minimum of computational effort. Next, higher-resolution versions of the pattern and image are used to refine the position estimates. Computation is kept to a minimum by restricting the search to neighborhoods of the points identified at the coarser resolution.

Signal enhancement

Signal enhancement is another area where MR decompositions can be used to reduce random noise in a degraded signal while sharpening details of the signal itself. The underlying idea is that real-world signals and noise possess rather distinct properties in the transform domain. The detail coefficients in each level of the MR representation are passed through some kind of thresholding function where small values (which are likely to include most of the noise) are set to zero, while larger values (which include prominent signal features) are retained. The final enhanced signal is obtained by reconstructing the levels of the processed MR representation.

Image fusion

Image fusion is a methodology concerned with the integration of multiple images, e.g. derived from different sensors, into a composite image that is more suitable for the purposes of human visual perception or computer-processing tasks. Since the essential goal of fusion is to preserve image features from the sources, a plausible approach is to transform the images into representations that decompose the images into relevant features such as edges, and perform fusion in this domain. An MR representation facilitates this type of analysis because it decomposes an image into different scales while preserving locality in space. MR image fusion is one of the main topics of this thesis and will be discussed in great detail.

Progressive transmission

For many applications, such as web browsing, it is desirable that a low-resolution version image can be made available very fast, and that further refinements eventually resulting in a high-resolution image, become available as time goes on. This is known as progressive transmission. Although other non-multiresolution image coding techniques can be modified to allow progressive transmission, MR representations are inherently suitable for this purpose by simply sending information from successive levels of the MR representation.

The above list of applications is far from exhaustive. The idea of representing a signal at multiple resolutions is used in several applications in signal processing and computer vision: speech recognition, texture classification, edge detection, image segmentation, surface reconstruction, image registration, motion analysis and optical flow estimation, to name only a few.

1.3 A case study: the Burt-Adelson pyramid

One of the earliest MR approaches in image processing is the *pyramid* representation proposed by Burt and Adelson [19][4]. A classical image pyramid consists of a sequence of versions of an original image in which resolution is gradually decreased by filtering and subsampling. The bottom (or zero) level x^0 of the pyramid is equal to the original image x. This image is low-pass filtered and subsampled to obtain the next level x^1, which is then filtered and subsampled in the same way to obtain x^2. Further repetitions of this filtering/subsampling procedure generate the subsequent levels of the pyramid, also known as a *low-pass pyramid.*

To illustrate the idea, consider a discrete image $x\colon \mathbb{Z}^2 \to \mathbb{R}$ and let $x^0 = x$. Each successive level image x^{k+1}, $k \geq 0$, is constructed from its lower level by

$$x^{k+1}(m,n) = \sum_{i\in\mathbb{Z}} \sum_{j\in\mathbb{Z}} w(i,j) x^k(2m-i, 2n-j),$$

where $w\colon \mathbb{Z}^2 \to \mathbb{R}$ are the filter coefficients. For simplicity, assume that w is separable, namely, $w(m,n) = h(m)h(n)$ where the filter h has length $2N+1$. If h is chosen properly, then the representation at each successive level will correspond to coarser and coarser structures in the original image. Burt and Adelson proposed the following design criteria for selecting the filter coefficients [19]:

(*i*) normalization: $\sum_{i=-N}^{N} h(i) = 1$;

(*ii*) symmetry: $h(i) = h(-i)$ for all i;

(*iii*) equal contribution: $\sum_{i=-N}^{N} h(2i) = \sum_{i=-N}^{N} h(2i+1)$, that is, all samples in a given level contribute equally to the next higher level.

A larger length of the support of h increases the number of degrees of freedom in the design, at the cost of increased computational cost. Burt and Adelson proposed a length of 5. Then, conditions $(i) - (iii)$ imply that the filter h has to be of the form:

$$h(0) = a, \qquad h(-1) = h(1) = \frac{1}{4}, \qquad h(-2) = h(2) = \frac{1}{4} - \frac{a}{2}. \tag{1.1}$$

If $a = 0.4$, the shape of the corresponding smoothing filter w resembles a Gaussian function, hence they referred to the corresponding low-pass pyramid as the *Gaussian pyramid.*

By interpolating each image x^{k+1} of the Gaussian pyramid and subtracting it from its predecessor x^k, one obtains the *Laplacian pyramid.* In [19], the interpolation operation is defined as

$$\hat{x}^k(m,n) = 4 \sum_{i=-N}^{N} \sum_{j=-N}^{N} w(i,j) x^{k+1}\left(\frac{m-i}{2}, \frac{n-j}{2}\right),$$

---

[4] In fact, an earlier pyramid representation had already been introduced, almost simultaneously, by Burt [16] and Crowley [39].

where only the terms for which $m-i$ and $n-j$ are even are to be included in the sum. The image $\hat{x}^k$ can be interpreted as a prediction of x^k. Thus, each detail image[5] y^{k+1} of the Laplacian pyramid corresponds to the error of approximating x^k by $\hat{x}^k$, i.e.,

$$y^{k+1}(m,n) = x^k(m,n) - \hat{x}^k(m,n)\,, \quad 0 \le k < K\,,$$

and $y^{K+1}(m,n) = x^K(m,n)$, where x^K is the coarsest image in the Gaussian pyramid.

Note that the original signal x can be recovered exactly from the Laplacian pyramid. The image y^{K+1} is interpolated and added to y^K to form x^{K-1}, which is then interpolated and added to y^{K-1} to recover x^{K-2}, and so on until x^0 is reached. Note also that the representation of x^0 in terms of its decomposed images $\{y^1, \ldots, y^K,\ x^K\}$ is redundant in the sense that it produces more samples than are actually needed for representing x^0. Hence, the Laplacian pyramid is an *overcomplete* representation, i.e., it has both perfect reconstruction (completeness) and redundancy properties.

Fig. 1.4 shows an example of the Gaussian as well as the Laplacian pyramid. The Gaussian pyramid has been generated with a filter of the form given by (1.1) with $a = 0.35$.

In the next chapter, where we propose an axiomatic framework for MR decompositions, we will revisit the Burt-Adelson pyramid.

1.4 Wavelets

In this section we introduce the basics of wavelet theory. For a more exhaustive study we refer to [44, 99, 107, 143, 163].

1.4.1 The continuous wavelet transform

Traditionally, a wavelet is a function $\psi \in L^2(\mathbb{R})$ with zero average:

$$\int_{-\infty}^{\infty} \psi(t)dt = 0\,.$$

The wavelet transform decomposes a signal over a family of wavelet functions obtained by translates and dilates of ψ, i.e.,

$$\psi_{a,b}(t) = |a|^{-1/2}\psi\left(\frac{t-b}{a}\right) \qquad a, b \in \mathbb{R}\,, a \neq 0\,.$$

For simplicity, we assume that ψ is a real function in the time domain centered at $t = 0$ and normalized so that $\|\psi\| = 1$. The *continuous wavelet transform* of $f \in L^2(\mathbb{R})$ at scale a and location b is computed by correlating f with $\psi_{a,b}$:

$$\mathcal{W}f(a,b) = \langle f, \psi_{a,b} \rangle = |a|^{-1/2} \int_{-\infty}^{\infty} f(t)\psi\left(\frac{t-b}{a}\right)\, dt\,.$$

---

[5] The indexing of levels that we use for the Laplacian pyramid is different from the one proposed by Burt and Adelson. We index the different levels from $k = 1$ to $K+1$, while Burt and Adelson index them from $k = 0$ to $k = K$. Our indexing is consistent with the notation introduced in Chapter 2 for the general framework of MR decompositions.

Figure 1.4: *Example of a Gaussian (left) and a Laplacian (right plus top-left image) pyramid. For the Gaussian pyramid (from bottom to top) images* x^0, x^1, x^2 *and* x^3 *are depicted. For the Laplacian (from bottom to top) images* y^1, y^2, y^3, $y^4 = x^3$ *are shown. The coarse approximation image* x^3 *in combination with the detail images* y^1, y^2, y^3 *provide an alternative (but redundant) representation of the original image* x^0.

As seen in the above equation, the transform $\mathcal{W}f$ is a continuous function of b and a, the translation and scale parameters, respectively. In the time domain, the wavelet $\psi_{a,b}$ is centered at b with a spread proportional to a. As scale increases, the wavelet becomes wider in time and hence narrower in the frequency domain (i.e., time resolution decreases but the frequency resolution increases). The joint resolution of $\psi_{a,b}$ can be represented in the time-frequency plane (t, w) by a tile as explained in Section 1.1.2. Here, because $\psi_{a,b}$ is centered at b and has unit energy, we have that

$$u = \frac{1}{2\pi}\int_{-\infty}^{\infty} w|\Psi_{a,b}(w)|^2 dw\,,$$

and

$$\sigma_t^2 = \int_{-\infty}^{\infty}(t-b)^2|\psi(t)|^2 dt\,, \qquad \sigma_w^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty}(w-u)^2|\Psi_{a,b}(w)|^2 dw\,,$$

where $\Psi_{a,b}$ is the Fourier transform of $\psi_{a,b}$. Thus, the tile would be centered at $(b, u/a)$ with size $a\sigma_t$ along time and σ_w/a along frequency. The area of the tile remains equal to $\sigma_t\sigma_w$ (which is at least $1/2$ by the uncertainty principle) at all scales but the resolution in time and frequency depends on a.

Since ψ has a zero average, $\mathcal{W}f(a,b)$ measures the variation of f in a neighborhood of b (location) whose size is proportional to a (scale). Sharp transitions in f at different locations and scales create corresponding large amplitude wavelet transform values.

The function f can be recovered from its wavelet transform provided that ψ satisfies the admissibility condition [99]:

$$\int_0^{\infty}\frac{|\Psi(w)|^2}{w}dw < \infty\,.$$

The continuous wavelet transform is redundant since it maps a one-dimensional signal onto a two-dimensional function. This redundancy can be reduced and even removed by subsampling the scale and translation parameters. A common choice is

$$a = a_0^k\,,\ b = nb_0a_0^k \qquad a_0 > 1, b_0 > 0 \quad k, n \in \mathbb{Z}\,.$$

The discretized family of wavelets is now of the form $\psi_{k,n}(t) = a_0^{-k/2}\psi(a_0^{-k}t - nb_0)$.

It has been shown [44] that stable reconstruction of $f \in L^2(\mathbb{R})$ from the wavelet coefficients $\langle f, \psi_{k,n}\rangle$, $k, n \in \mathbb{Z}$, is possible if and only if

$$A\|f\|^2 \leq \sum_{k=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}|\langle f, \psi_{k,n}\rangle|^2 \leq B\|f\|^2\,,$$

for certain constants $0 < A \leq B < \infty$. If this condition is satisfied, then the family $\{\psi_{k,n}\}_{k,n\in\mathbb{Z}}$ constitutes a *frame.* In [44], necessary and sufficient conditions on ψ, a_0 and b_0 are given so that $\{\psi_{k,n}\}_{k,n\in\mathbb{Z}}$ is a frame of $L^2(\mathbb{R})$. In this case, there exist a *dual frame* $\{\tilde{\psi}_{k,n}\}_{k,n\in\mathbb{Z}}$ such that f can be reconstructed by

$$f(t) = \sum_{k=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\langle f, \psi_{k,n}\rangle\tilde{\psi}_{k,n}(t)\,. \tag{1.2}$$

From frame theory, we know that

$$\tilde{\psi}_{k,n} = (U^*U)^{-1}\psi_{k,n}\,, \tag{1.3}$$

where U is the corresponding frame operator[6] (i.e., $Uf(k,n) = \langle f, \psi_{k,n}\rangle$) and U^* its adjoint. Contrary to what one might expect, the dual function $\tilde{\psi}_{k,n}$ is, generally, *not* obtained by scaling and translating a dual wavelet $\tilde{\psi}$. However, it has been shown [99] that

$$\tilde{\psi}_{k,n}(t) = \tilde{\psi}_{k,0}(t - na_0^k b_0)\,, \tag{1.4}$$

which means that $\tilde{\psi}_{k,n}$ can be obtained by computing each $\tilde{\psi}_{k,0}$ with (1.3) and translating it according to (1.4).

1.4.2 Mallat's multiresolution analysis

In general, the expansion in (1.2) is still an overcomplete representation of f. As a matter of fact, the wavelet series $\{\langle f, \psi_{k,n}\rangle\}_{k,n\in\mathbb{Z}}$ is simply a sampled version of the continuous wavelet transform. Eliminating the redundancy is equivalent to building a basis of $L^2(\mathbb{R})$. A theoretical framework for constructing wavelet bases is *multiresolution analysis*[7] [97, 107].

The idea behind multiresolution analysis is to compute the approximation of signals at various scales or resolutions by defining appropriate projections onto different spaces V_k. Not surprisingly, it was inspired by the pyramid representation of Burt and Adelson [19], and, as we will show later, it provides a connection with perfect reconstruction filter banks. The following definition specifies the mathematical properties of multiresolution analysis spaces for a dyadic scale progression, i.e., $a = 2^k$, $k \in \mathbb{Z}$.

Definition 1.4.1. A sequence $\{V_k\}_{k\in\mathbb{Z}}$ of closed subspaces of $L^2(\mathbb{R})$ is called a *multiresolution analysis* of $L^2(\mathbb{R})$ if there exists a function $\phi \in V_0$, with a non-vanishing integral, such that $\{\phi(\cdot - n)\}_{n\in\mathbb{Z}}$ is a Riesz basis of V_0, and

$$V_{k+1} \subset V_k\,, \qquad k \in \mathbb{Z} \tag{1.5}$$

$$f \in V_k \iff f(\cdot - 2^k n) \in V_k\,, \qquad k, n \in \mathbb{Z} \tag{1.6}$$

$$f \in V_k \iff f(\cdot/2) \in V_{k+1}\,, \qquad k \in \mathbb{Z} \tag{1.7}$$

$$\bigcap_{k=-\infty}^{\infty} V_k = \{\mathbf{0}\}\,, \tag{1.8}$$

$$\overline{\bigcup_{k=-\infty}^{\infty} V_k} = L^2(\mathbb{R})\,, \tag{1.9}$$

where $\overline{S}$ denotes the closure of set S, and $\mathbf{0}$ denotes the signal which is identically zero. The function ϕ is called the *scaling function*.

---

[6] The frame operator U: $L^2(\mathbb{R}) \to l^2(J)$ associated with a frame $\{\gamma_j\}_{j\in J}$, is given by $Uf(j) = \langle f, \gamma_j\rangle$. The adjoint of U is another operator U^*: $l^2(J) \to L^2(\mathbb{R})$ such that $\langle Uf, x\rangle = \langle f, U^*x\rangle$, $x \in l^2(J)$.

[7] Most wavelet bases can be derived from multiresolution analysis but not all of them.

From this definition, it is obvious that $\{\phi_{k,n}\}_{n\in\mathbb{Z}}$, where $\phi_{k,n}(t) = 2^{-k/2}\phi(2^{-k}t-n)$, is a Riesz basis of V_k. Since $V_k \subset V_{k-1}$, any element of V_k can be obtained as a linear combination of basis functions of V_{k-1}. In particular, $\phi \in V_0 \subset V_{-1}$ can be expressed as

$$\phi(t) = 2^{1/2} \sum_{n=-\infty}^{\infty} h(n)\phi(2t-n)\,.$$

The sequence $\{h(n)\}_{n\in\mathbb{Z}} \in l^2(\mathbb{Z})$ entirely determines the scaling function ϕ and the corresponding multiresolution analysis. Furthermore, ϕ will be compactly supported only if the support of h is finite.

We now explain how multiresolution analysis theory can be used for the construction of wavelets. We start by introducing the 'detail space' W_k containing the detail information needed to go from an approximation at V_k to a finer approximation at V_{k-1}.

Let W_k be the space complementing V_k in V_{k-1}, that is,

$$V_{k-1} = V_k \oplus W_k\,, \tag{1.10}$$

where $\oplus$ stands for the direct sum. A function ψ is called a wavelet if $\{\psi(\cdot - n)\}_{n\in\mathbb{Z}}$ is a Riesz basis of W_0. From the multiresolution analysis definition and (1.10), it follows that the family $\{\psi_{k,n}\}_{k,n\in\mathbb{Z}}$, where $\psi_{k,n}(t) = 2^{-k/2}\psi(2^{-k}t-n)$, constitutes a Riesz basis of $L^2(\mathbb{R})$.

Since $\psi \in W_0 \subset V_{-1}$ and $\{\phi_{k,n}\}_{n\in\mathbb{Z}}$ is a Riesz basis of V_k, the wavelet ψ can be expressed as

$$\psi(t) = 2^{1/2} \sum_{n=-\infty}^{\infty} g(n)\phi(2t-n)\,,$$

where $\{g(n)\}_{n\in\mathbb{Z}} \in l^2(\mathbb{Z})$.

At this point, the exact conditions on the scaling and wavelet functions, or alternatively, on the sequences h and g, are not given yet. They will be given below and in the next subsections.

Let $\{V_k\}_{k\in\mathbb{Z}}$, $\{\tilde{V}_k\}_{k\in\mathbb{Z}}$ be two multiresolution analyses of $L^2(\mathbb{R})$ with scaling functions ϕ, $\tilde{\phi}$, respectively. The construction of biorthogonal wavelets $\psi \in W_0$, $\tilde{\psi} \in \tilde{W}_0$ imposes the following requirements:

$$\langle\tilde{\phi}, \phi(\cdot - n)\rangle \;= \langle\tilde{\psi}, \psi(\cdot - n)\rangle \;= \delta(n) \tag{1.11}$$

$$\langle\tilde{\phi}, \psi(\cdot - n)\rangle \;= \langle\tilde{\psi}, \phi(\cdot - n)\rangle \;= 0\,, \tag{1.12}$$

which are also known as *biorthogonality conditions*. The two multiresolution analyses may also coincide. For the particular case where $\psi = \tilde{\psi}$, we have that ψ is an *orthogonal wavelet* (i.e., $\{\psi_{k,n}\}_{k,n\in\mathbb{Z}}$ is an orthogonal basis of $L^2(\mathbb{R})$).

1.4.3 Wavelets and filter banks

We now explain how multiresolution analysis theory is related to perfect reconstruction filter banks. Recall that

$$\phi(t) = 2^{1/2} \sum_{n=-\infty}^{\infty} h(n)\phi(2t-n) \quad \text{and} \quad \psi(t) = 2^{1/2} \sum_{n=-\infty}^{\infty} g(n)\phi(2t-n)\,.$$

We refer to these relations as the *refinement equations*. As we will show below, h corresponds to the impulse response of a low-pass synthesis filter while g corresponds to the impulse response of a high-pass synthesis filter. Obviously, the refinement equations also hold for the dual functions $\tilde{\phi}$, $\tilde{\psi}$, yielding the dual discrete filters $\tilde{h}$ and $\tilde{g}$, respectively.

Consider now a signal $f \in V_k$. Then, f can be written as a linear combination of $\phi_{k,n}$, $n \in \mathbb{Z}$, i.e.,

$$f(t) = \sum_{n=-\infty}^{\infty} x^k(n)\phi_{k,n}(t)\,,$$

where $x^k(n) = \langle f, \tilde{\phi}_{k,n}\rangle$. We interpret x^k as the discrete approximation of f. Since $V_k = V_{k+1} \oplus W_{k+1}$, we may also write

$$f(t) = \sum_{n=-\infty}^{\infty} x^{k+1}(n)\phi_{k+1,n}(t) + \sum_{n=-\infty}^{\infty} y^{k+1}(n)\psi_{k+1,n}(t)\,,$$

where $x^{k+1}(n) = \langle f, \tilde{\phi}_{k+1,n}\rangle$ and $y^{k+1}(n) = \langle f, \tilde{\psi}_{k+1,n}\rangle$. The coefficients $x^{k+1}(n)$, $n \in \mathbb{Z}$, are the *scaling* or *approximation* coefficients, while $y^{k+1}(n)$, $n \in \mathbb{Z}$, are the *wavelet* or *detail* coefficients.

Since $\{\psi_{k,n}\}_{k,n\in\mathbb{Z}}$ is a basis of $L^2(\mathbb{R})$, any signal $f \in L^2(\mathbb{R})$ can be written as

$$f(t) = \sum_{k=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} y^k(n)\psi_{k,n}(t)\,,$$

and also as

$$f(t) = \sum_{n=-\infty}^{\infty} x^K(n)\phi_{K,n}(t) + \sum_{k=-\infty}^{K}\sum_{n=-\infty}^{\infty} y^k(n)\psi_{k,n}(t)\,.$$

We refer to either of the above representations as the *wavelet decomposition* of f. The transformation that maps a continuous signal f onto the discrete signals x^K, $\{y^k\}_{k\leq K}$ or $\{y^k\}_{k\in\mathbb{Z}}$ is referred to as the *wavelet transform* of f.

Using the refinement equations, it is easy to verify that x^{k+1} and y^{k+1} can be found by

$$x^{k+1}(n) = \sum_{l=-\infty}^{\infty} \tilde{h}(l-2n)x^k(l)\,, \quad y^{k+1}(n) = \sum_{l=-\infty}^{\infty} \tilde{g}(l-2n)x^k(l)\,. \tag{1.13}$$

This means in particular that there is no need to calculate the inner products explicitly. A similar computation shows that the inverse transformation is given by

$$x^k(n) = \sum_{l=-\infty}^{\infty} h(n-2l)x^{k+1}(l) + \sum_{l=-\infty}^{\infty} g(n-2l)y^{k+1}(l)\,. \tag{1.14}$$

The expressions in (1.13)-(1.14) are in conformity with the two-channel perfect reconstruction filter bank depicted in Fig. 1.5, where the analysis filters $\bar{h}$, $\bar{g}$ are defined as $\bar{h}(n) = \tilde{h}(-n)$, $\bar{g}(n) = \tilde{g}(-n)$.

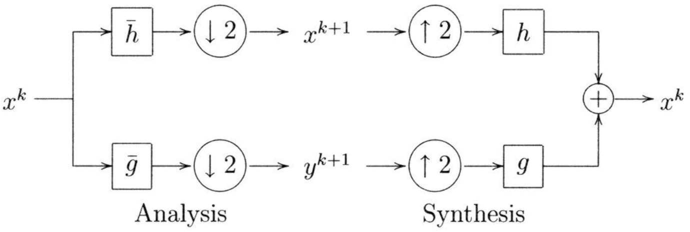


Figure 1.5: *Perfect reconstruction filter bank with low-pass and high-pass analysis filters* $\bar{h}$, $\bar{g}$, *respectively, and low-pass and high-pass synthesis filters* h, g, *respectively.*

Here, the input signal x^k is decomposed into an approximation signal x^{k+1} and a detail signal y^{k+1} by filtering with $\bar{h}$ and $\bar{g}$, respectively, and downsampling:

$$x^{k+1}(n) = (\bar{h} * x^k)(2n), \quad y^{k+1}(n) = (\bar{g} * x^k)(2n),$$

which is equivalent to (1.13). Synthesis is achieved by upsampling signals x^{k+1} and y^{k+1}, filtering with h and g, respectively, and adding the respective outputs. Thus,

$$x^k(n) = (h * \breve{x}^{k+1})(n) + (g * \breve{y}^{k+1})(n),$$

where $\breve{x}$ denotes the upsampling of x, i.e., $\breve{x}(2n) = x(n)$ and $\breve{x}(2n+1) = 0$. Hence, the above expression is equivalent to (1.14).

1.4.4 How to choose a wavelet?

For perfect reconstruction, the analysis and synthesis filters need to satisfy specific constraints [99, 163]:

$$\sum_{l=-\infty}^{\infty} h(l)\bar{h}(2n-l) = \sum_{l=-\infty}^{\infty} g(l)\bar{g}(2n-l) \quad = \quad \delta(n) \tag{1.15}$$

$$\sum_{l=-\infty}^{\infty} h(l)\bar{g}(2n-l) = \sum_{l=-\infty}^{\infty} g(l)\bar{h}(2n-l) \quad = \quad 0\,. \tag{1.16}$$

One can easily establish the equivalence between the biorthogonality conditions stated in (1.11)-(1.12) and (1.15)-(1.16). Under some additional conditions on the filters, the discrete wavelet transform described above is orthonormal. Orthonormality implies that the energy of the signal is preserved under transformation. If these conditions are met, the synthesis filters are a reflected version of the analysis filters, and the high-pass filters are modulated versions of the low-pass filters, namely,

$$h(n) = \bar{h}(-n), \quad g(n) = \bar{g}(-n), \quad \bar{g}(n) = (-1)^n \bar{h}(M-n)$$

where M is an integer delay. Such filters are often known as *conjugate mirror filters*.

Therefore, the wavelet decomposition (and reconstruction) of a discrete signal from a resolution to the next one is implemented by a two-channel perfect reconstruction filter bank. By iterating on the approximation signal, filter banks can be used to obtain biorthogonal wavelet bases of $L^2(\mathbb{R})$, assuming that the filters satisfy some stability constraints [99].

There exists a great variety of wavelet families depending on the choice of the prototype wavelet or, alternatively, the filter's coefficients. However, imposing additional requirements such as orthogonality, symmetry, compactness of support, rapid decay and smoothness limits our choice. The 'optimal' choice of the wavelet basis will depend on the application at hand, and therein lies part of the difficulty of building a suitable wavelet representation.

One of the most useful properties of wavelets are vanishing moments. A wavelet ψ is said to have p vanishing moments if

$$\int_{-\infty}^{\infty} t^k \psi(t)dt = 0 \qquad \text{for } 0 \leq k < p.$$

This property results in a sparse representation of piecewise smooth signals because the wavelet coefficients will be essentially zero over all regions where the signal is well approximated by a polynomial.

Intuitively speaking, wavelets are successful in many signal applications because their decorrelation potential leads to sparse representations and their scaling property allows them to 'zoom in' on singularities as well as to support MR representation.

One drawback of the wavelet transform and, to a lesser extent, also for the pyramid transform, is that it generally yields a shift-variant signal representation. This means that a simple shift of the input signal may lead to complete different transform coefficients. The lack of translation invariance can be avoided if the outputs of the filter banks are not decimated. The resulting undecimated wavelet transform [99] yields a redundant MR representation where the approximation and detail signals have all the same size as the original signal.

Most wavelet (and pyramid) transforms have been designed in the one-dimensional case. By successive application of such one-dimensional transforms on the rows and the columns (or vice versa) of an image, one obtains a so-called *separable* two-dimensional transform. This construction is illustrated in Fig. 1.6 for the wavelet transform. At each level k, the input x^k is decomposed into a coarse approximation x^{k+1} and three detail signals $y^{k+1} = \{y^{k+1}(\cdot|1), y^{k+1}(\cdot|2), y^{k+1}(\cdot|3)\}$, corresponding to the horizontal, vertical and diagonal directions. Fig. 1.7 shows a two-level wavelet decomposition computed in this way.

Non-separable transforms can also be constructed [70, 85, 138]. Although they provide decompositions with more general properties, they have been used less often in image applications due to the lack of general tools for their design.

1.4.5 The need for adaptive wavelets

Wavelets have had a tremendous impact on signal processing, both because of their unifying role and their success in several applications. The applicability of the wavelet transform (as

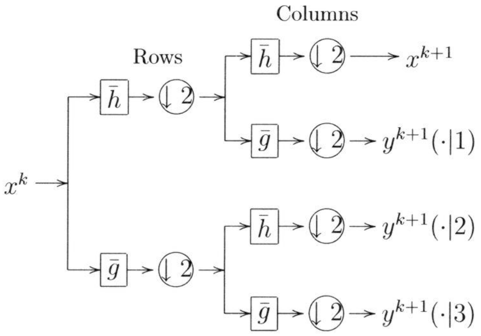


Figure 1.6: *Separable two-dimensional wavelet transform. Here* $\bar{h}$ *and* $\bar{g}$ *are the low-pass and high-pass analysis filters, respectively.*

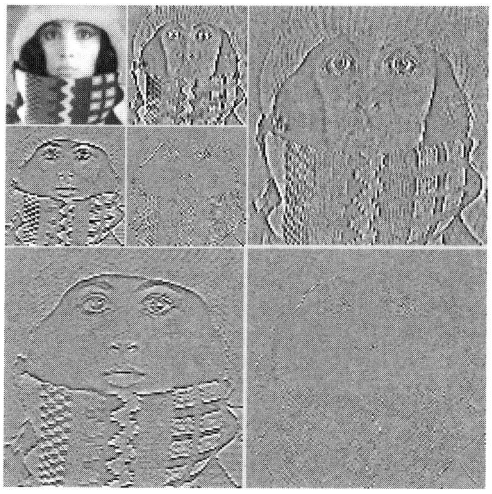


Figure 1.7: *Example of a two-level discrete wavelet transform. In the upper-left quarter, the second level is displayed. Starting from the top left and going clockwise: approximation, vertical, diagonal and horizontal detail images. The upper-right, bottom-right and bottom-left quarters show respectively the vertical, diagonal and horizontal first-level details.*

well as for other MR decompositions) is somewhat limited, however, by the linearity assumption. Coarsening a signal by means of linear operators may not be compatible with a natural coarsening of some signal attribute of interest (e.g., the shape of an object), and hence the use of linear procedures may be inconsistent in such applications. In general, linear filters smear the singularities of a signal and displaces their locations, causing undesirable effects.

Moreover, standard wavelets are often not suited for higher dimensional signals because they are not adapted to the 'geometry' of higher dimensional signal singularities. For example, an image comprises smooth regions separated by piecewise regular curves. Wavelets, however, are good at isolating the discontinuity across the curve, but they do not 'see' the smoothness along the curve. These observations indicate the need for MR representations which are data-dependent.

In this thesis, we will look for adaptive (i.e., data-driven) transforms which retain the desirable properties of the standard wavelet transform (e.g., non-redundancy and invertibility) while exploiting, in a simple way, the geometrical information of the underlying signal. This will allow for a better localization and representation of the singularities, as well as for sharper (perceptually better) approximations at lower resolutions.

1.5 Application to image fusion

During the past decade, several image fusion algorithms have been developed and presented in the literature. Many of them are based on MR techniques of image processing. It has been shown that these methods exhibit a high level of image quality and a good degree of robustness.

Image fusion based on MR approaches is motivated by the fact that the human visual system is primarily sensitive to local contrast changes, i.e. edges, and MR decompositions provide a convenient space-scale localization of these local changes. In addition, MR techniques are also very convenient for related applications such as image registration and spatial enhancement.

The basic strategy of a generic MR image fusion scheme is to use specific fusion rules to construct a composite MR representation from the MR representations of the different input images. The fused image is then obtained by applying the inverse decomposition process.

Fusion using MR decompositions will be described comprehensively in Chapter 6 and 7.

1.6 Outline of the thesis

Chapter 2 presents an axiomatic framework encompassing most existing linear and nonlinear MR decompositions. Chapter 3 is concerned with the construction of adaptive wavelets. In particular, we study a new method for the construction of adaptive wavelets using update lifting. The key ingredient of our method is a binary decision map that triggers the choice of the update filters. This decision map is obtained by thresholding the size of the gradient resulting from some seminorm. Moreover, we establish conditions under which the decision map can be recovered at synthesis, without the need for transmitting overhead information. Chapter 4 treats various particular cases. Several simulations examples are shown both for one and two-dimensional signals. Chapter 5 focuses on one of the main applications of wavelets, namely image compression. We analyze in particular the quantization effects in our adaptive lifting scheme. We derive sufficient conditions for recovering the original decisions and show how the reconstruction error can be bounded by the quantization error. The remainder of the thesis is devoted to yet another application: image fusion. Chapter 6 introduces the concept of image fusion and presents a general framework for multiresolution image fusion. We also illustrate how adaptive decompositions can be used in such a framework. In Chapter 6, the actual fusion is done at pixel level. If fusion is preceded by segmentation of the source images, then it is possible to perform the actual fusion at the region level. This is the topic of Chapter 7. Chapter 8 addresses the topic of performance assessment in image fusion and proposes a new objective non-reference measure for such a task. Finally, in Chapter 9 we put our results in a broad perspective, draw some conclusions and discuss current and future work. A Matlab toolbox[8] which implements both pixel and region-based image fusion approaches has been developed. A brief overview of this toolbox is given in Appendix A.

---

[8] MATIFUS: MAtlab Toolbox for Image FUSion.

Chapter 2

Multiresolution decomposition systems

A multiresolution decomposition scheme decomposes the signal being analyzed into several components, each of which captures information present at a given level of resolution. The notion of resolution relates to the size of the details that can be represented. As observed in Chapter 1, multiresolution approaches are very useful in signal processing for various reasons, e.g. : (*i*) signals usually consist of structures at different scales; (*ii*) there is strong evidence that the human perceptual and visual systems process information in a multiresolution fashion; (*iii*) multiresolution methods offer computational advantages and, moreover, appear to be robust.

In this chapter we study in more detail multiresolution decomposition systems. In Section 2.1, we discuss the general concept of a decomposition system with perfect reconstruction and we explain how concatenation of such systems can lead to multiresolution decompositions. In Section 2.2 we introduce a general method, called lifting, which can be exploited to modify a given decomposition system. While there exist several types of multiresolution decompositions, we mainly concentrate on two well-known special classes: pyramids and wavelets, which are discussed in Section 2.3 and Section 2.4, respectively. They are studied within the axiomatic framework proposed by Heijmans and Goutsias in [63, 70]. Finally, we give a brief description of other multiresolution decomposition systems in Section 2.5.

The contributions made in this chapter can be grouped in two broad categories. First, we review existing multiresolution decomposition systems within the general framework proposed in [63, 70]. Second, we extend the lifting scheme (usually restricted to wavelets) to general decompositions schemes.

2.1 Decomposition systems with perfect reconstruction

When analyzing a signal, it is often useful to decompose it into different parts. Those parts can then be analyzed separately which may facilitate subsequent processing tasks. Of particular interest is the case where the signal is decomposed in such a way that no information is removed and the original signal can be recovered exactly (*perfect reconstruction*) from its constituting parts.

The idea of a decomposition system with perfect reconstruction is to obtain a more convenient representation (*analysis*) of the signal such that no information is lost, i.e., the signal can be recovered through some reconstruction process (*synthesis*). Fig. 2.1 depicts a general scheme for the decomposition of an input signal $x^0 \in V_0$ into two components $(x^1, y^1) \in V_1 \times W_1$. Here, x^1 and y^1 can be interpreted as the *approximation* and *detail* signals of x^0, respectively. In other words, x^1 is a sort of 'simplification' of x^0, inheriting many of its properties, whereas y^1 is a kind of 'refinement' that contains the information that has been discarded in the simplification process. The operators $\psi^\uparrow\colon V_0 \to V_1$, $\omega^\uparrow\colon V_0 \to W_1$ are called *analysis operators* and the operator $\Psi^\downarrow\colon V_1 \times W_1 \to V_0$ is called the *synthesis operator*. The assumption that no information is lost by the decomposition is expressed by the requirement that $\Psi^\downarrow$ is the left inverse of $\Psi^\uparrow = (\psi^\uparrow, \omega^\uparrow)$, i.e.,

$$\Psi^\downarrow\left(\psi^\uparrow(x^0), \omega^\uparrow(x^0)\right) = x^0\,, \quad \text{for } x^0 \in V_0\,.$$

This condition will be referred to as the *perfect reconstruction condition*.

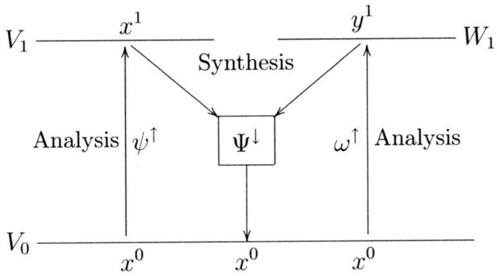


Figure 2.1: *A signal decomposition scheme with perfect reconstruction.*

In various signal and image applications, the decomposition $x^0 \mapsto (x^1, y^1)$ is only a first step toward an analysis of x^0. Subsequent steps comprise a decomposition of x^1 into x^2 and y^2, of x^2 into x^3 and y^3, and so forth. By concatenating several systems of the form depicted in Fig. 2.1 we obtain a *multilevel decomposition system*. If the higher levels are obtained by means of some filtering (e.g., linear or morphological) of the lower level signals, possibly followed by a sampling step, then we call the system a *multiresolution* decomposition scheme.

To formalize this procedure, assume that there exists a sequence of signal spaces V_k, $k \geq 0$, and detail spaces W_k, $k \geq 1$. At each level $k \geq 0$ we have two analysis operators, $\psi_k^\uparrow\colon V_k \to V_{k+1}$ and $\omega_k^\uparrow\colon V_k \to W_{k+1}$, and a synthesis operator $\Psi_k^\downarrow\colon V_{k+1} \times W_{k+1} \to V_k$, satisfying the perfect reconstruction condition:

$$\Psi_k^\downarrow\left(\psi_k^\uparrow(x), \omega_k^\uparrow(x)\right) = x, \quad \text{for } x \in V_k\,. \tag{2.1}$$

A given input signal $x^0 \in V_0$ can be decomposed by the recursive scheme

$$x^0 \to \{y^1, x^1\} \to \{y^1, y^2, x^2\} \to \cdots \to \{y^1, \ldots, y^{K-1}, y^K, x^K\}\,, \tag{2.2}$$

where

$$\begin{cases} x^{k+1} = \psi_k^{\uparrow}(x^k) \\ y^{k+1} = \omega_k^{\uparrow}(x^k) \end{cases} \qquad k = 0, 1, \ldots, K-1\,. \tag{2.3}$$

Here, x^{k+1} is an approximation of x^k, but can also be regarded as a $(k+1)$'th-order coarse approximation of the original signal x^0. In contrast, the detail signal y^{k+1} contains information about x^k that is not present in the simplified component x^{k+1}.

Note that, because of the perfect reconstruction condition, the original signal x^0 can be perfectly reconstructed from x^K and $y^1, y^2, \ldots, y^K$ by means of the backward recursion:

$$x^k = \Psi_k^{\downarrow}(x^{k+1}, y^{k+1}), \qquad k = K-1, K-2, \ldots, 0\,. \tag{2.4}$$

Fig. 2.2 illustrates the analysis and synthesis schemes for the particular case where $K=3$.

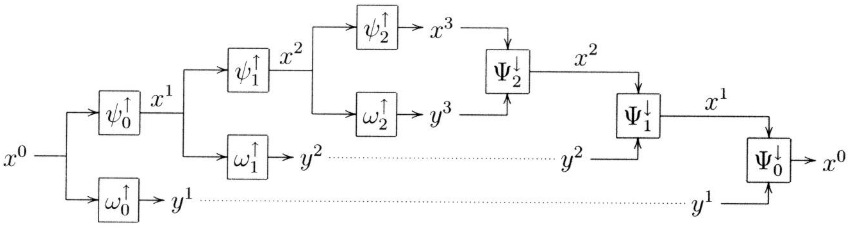


Figure 2.2: *A 3-level decomposition system: analysis and synthesis.*

2.2 Lifting

In this section we describe a general and flexible technique to modify a given decomposition system into another one, possibly with some improved characteristics. This technique, called *lifting*, was developed by Sweldens in the context of wavelets [45, 146, 147] (see also [14] for a related scheme, known as a *ladder network*), but it can be applied to general decompositions systems.

A general lifting scheme starts with an invertible decomposition of the input signal $x^0 \in V_0$ into two parts, the approximation signal $x \in V_1$, and the detail signal $y \in W_1$. Two types of lifting schemes can be distinguished: *prediction lifting* and *update lifting*. We treat these cases separately. In both cases, we consider an initial signal decomposition scheme with perfect reconstruction such as depicted in Fig. 2.1, with analysis operators $\Psi^{\uparrow} = (\psi^{\uparrow}, \omega^{\uparrow})$ and a synthesis operator $\Psi^{\downarrow}$.

Prediction lifting

The detail signal y is predicted using information contained in the approximation signal x and is replaced by the prediction error

$$y' = y - P(x)\,,$$

where P: $V_1 \longrightarrow W_1$ represents the prediction operator. The prediction error y' becomes the new detail signal. In practice, the prediction operator P is chosen such that $P(x)$ is an estimate (i.e., a prediction) of y and hence the new signal y' is 'smaller' than y. Clearly, the original signal x^0 can be reconstructed from x and y' by

$$x^0 = \Psi^{\downarrow}(x, y) = \Psi^{\downarrow}\bigl(x, y' + P(x)\bigr) .$$

Thus, the prediction lifting yields a new perfect reconstruction decomposition system with analysis and synthesis operators given by

$$\psi_p^{\uparrow}(x) = \psi^{\uparrow}(x) , \qquad x \in V_0 \tag{2.5}$$

$$\omega_p^{\uparrow}(x) = \omega^{\uparrow}(x) - P\left(\psi^{\uparrow}(x)\right) , \qquad x \in V_0 \tag{2.6}$$

$$\Psi_p^{\downarrow}(x, y) = \Psi^{\downarrow}\bigl(x, y + P(x)\bigr) , \qquad x \in V_1 \, , y \in W_1 \, . \tag{2.7}$$

Update lifting

The approximation signal x is updated using information contained in the detail signal y:

$$x' = x + U(y) ,$$

where U: $W_1 \longrightarrow V_1$ represents the update operator. Generally, the update operator is chosen in such a way that the resulting signal x' satisfies a certain constraint. For example, one might require that the mapping $x^0 \longmapsto x'$ preserves a given signal attribute such as the average or some local maximum. As before, the original signal can be easily reconstructed from x' and y:

$$x^0 = \Psi^{\downarrow}(x, y) = \Psi^{\downarrow}(x' - U(y), y) .$$

Thus, the update lifting yields a new perfect reconstruction decomposition system with analysis and synthesis operators given by

$$\psi_u^{\uparrow}(x) = \psi^{\uparrow}(x) + U\left(\omega^{\uparrow}(x)\right) , \qquad x \in V_0 \tag{2.8}$$

$$\omega_u^{\uparrow}(x) = \omega^{\uparrow}(x) , \qquad x \in V_0 \tag{2.9}$$

$$\Psi_u^{\downarrow}(x, y) = \Psi^{\downarrow}\left(x - U(y), y\right) , \qquad x \in V_1 \, , y \in W_1 \, . \tag{2.10}$$

Therefore, an existing decomposition system with perfect reconstruction can be modified by an arbitrary prediction or update lifting step. Perfect reconstruction is guaranteed by the intrinsic structure of this scheme and does not require any particular assumptions on the lifting operators P and U. Moreover, the operators '+' and '−' used in the above expressions can be replaced by any pair of invertible operators. This flexibility has challenged researchers to develop various nonlinear transforms [22, 30, 148], including morphological ones [70].

Fig. 2.3 illustrates a prediction-update lifting scheme. At analysis, the prediction operator P acting on x is used to modify y, resulting in a new detail signal y'. Subsequently, the

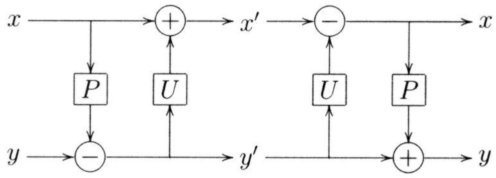


Figure 2.3: *Classical lifting scheme: analysis and synthesis.*

update operator U acting on y' is used to modify x, yielding a new approximation signal x'. At synthesis, the signals x and y are reconstructed by reversing the lifting steps.

Obviously, we can concatenate any number and type of lifting steps in order to modify a given decomposition system. 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 transform may have more vanishing moments than the original one, or it may be superior in decorrelating signals within a given class. Another important feature of the lifting scheme not mentioned so far is that it allows in-place calculations. This means that the original signal values can be replaced by the transformed ones without having to allocate additional memory.

2.3 The pyramid transform

2.3.1 Axiomatics

The pyramid transform is a special case of a multiresolution decomposition with perfect reconstruction. It is characterized by the assumption that

$$\Psi_k^{\downarrow}(x,y) = \psi_k^{\downarrow}(x) + y, \quad \text{for } x \in V_{k+1}\,,\ y \in W_{k+1}\,, \tag{2.11}$$

where $W_{k+1} \subseteq V_k$ and $\psi_k^{\downarrow}\colon V_{k+1} \to V_k$. The perfect reconstruction condition in (2.1) can be reformulated as

$$\psi_k^{\downarrow}\psi_k^{\uparrow}(x) + \omega_k^{\uparrow}(x) = x, \quad \text{for } x \in V_k\,.$$

Thus, $\omega_k^{\uparrow}(x) = x - \psi_k^{\downarrow}\psi_k^{\uparrow}(x)$ is the error of the synthesis operator $\psi_k^{\downarrow}$ when reconstructing x from the approximation $\psi_k^{\uparrow}(x)$. In this case, the recursive analysis scheme in (2.2) is given by

$$\begin{cases} x^{k+1} = \psi_k^{\uparrow}(x^k) \\ y^{k+1} = x^k - \psi_k^{\downarrow}(x^{k+1}) \end{cases} \qquad k = 0, 1, \ldots, K-1\,, \tag{2.12}$$

and the synthesis step in (2.4) is

$$x^k = \psi_k^{\downarrow}(x^{k+1}) + y^{k+1}, \qquad k = K-1, K-2, \ldots, 0\,. \tag{2.13}$$

We refer to the decomposition process

$$x^0 \mapsto \{y^1, \dots, y^{K-1}, y^K, x^K\}$$

by means of (2.12) as the *pyramid transform* of x^0, and to the process of synthesizing x^0 by means of (2.13) as the *inverse pyramid transform*. A block diagram illustrating the pyramid transform and its inverse is shown in Fig. 2.4. We call the sequence $\{x^0, x^1, \dots, x^K\}$ the *approximation pyramid*, and the sequence $\{y^1, y^2, \dots, y^K\}$ the *detail pyramid*.

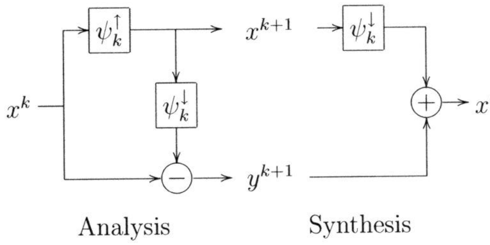


Figure 2.4: *Pyramid transform (analysis) and its inverse (synthesis).*

The axiomatic pyramid approach described above encompasses several existing pyramid techniques such as the well-known Laplacian pyramid introduced by Burt and Adelson in [19] and which has been described in Section 1.3.

Example 2.3.1 (Laplacian or Burt-Adelson pyramid). Let us consider that all spaces V_k are identical, namely $l^2(\mathbb{Z})$. Consider also that at every level k the same analysis and synthesis operators $\psi^\uparrow$, $\psi^\downarrow$ are used. In particular, let us choose $\psi^\uparrow$ as a linear filter followed by a dyadic downsampling, i.e.,

$$\psi^\uparrow(x)(n) = (\bar{h} * x)(2n) = \sum_{l=-\infty}^{\infty} \bar{h}(l)x(2n-l)$$

and $\psi^\downarrow$ as a dyadic upsampling followed by a linear filter, i.e.,

$$\psi^\downarrow(x)(n) = (h * \check{x})(n) = \sum_{l=-\infty}^{\infty} h(l)x(n-2l)\,.$$

Here, $\check{x}$ denotes the upsampling of x, i.e., $\check{x}(2n) = x(n)$ and $\check{x}(2n+1) = 0$, and $\bar{h}$, $h \in l^2(\mathbb{Z})$ are convolution kernels corresponding respectively to a smoothing and an interpolation filter. Choosing $h = 2\bar{h}$, we obtain the pyramid representation proposed by Burt and Adelson in [19] (see also Section 1.3). The sequence $\{x^k\}$, $k = 0, \dots, K$, of approximation signals corresponds to the Gaussian pyramid and the sequence $\{y^k\}$, $k = 1, \dots, K$, of detail signals along with x^K corresponds to the Laplacian pyramid. Fig. 1.4 shows an example of such pyramids. Here, $\bar{h}$ has length 5 and is of the form given by (1.1) with $a = 0.35$.

Obviously, the choice of different analysis and synthesis operators results in different kinds of pyramids. In particular, nonlinear pyramids have attracted a great deal of attention. A well-known example is the morphological pyramid [63, 153], where the analysis and synthesis operators are based on morphological operators such as erosions, dilations, openings and closings [131, 140]. Another instance of nonlinear pyramids is the ratio-of-low-pass pyramid [152]. Here, the ratio (rather than the standard difference) of the successive low-pass filtered signals is computed. In fact, the operator '+' used in (2.11) can be replaced by any invertible operation (see [63] for details).

Observe that a signal representation obtained by means of a pyramid transform (i.e., detail signals along with the coarsest approximation) is overcomplete in the sense that it produces more samples than the original signal. This is a direct consequence of the fact that the detail signal y^k 'lives' at the same resolution level as x^{k-1}.

Note also that we can go from any level k in the pyramid to a higher level l by successively composing analysis operators. This gives an operator

$$\psi^{\uparrow}_{k,l} = \psi^{\uparrow}_{l-1}\psi^{\uparrow}_{l-2}\cdots\psi^{\uparrow}_{k}\,, \quad l > k\,,$$

which maps a signal in V_k to a signal in V_l. On the other hand, the composed operator

$$\psi^{\downarrow}_{l,k} = \psi^{\downarrow}_{k}\psi^{\downarrow}_{k+1}\cdots\psi^{\downarrow}_{l-1}\,, \quad l > k\,,$$

takes a signal from level l to a lower level k. We define the *approximation operator*

$$\hat{\psi}_{k,l,=}\psi^{\downarrow}_{l,k}\psi^{\uparrow}_{k,l}\,, \quad l > k\,,$$

which takes a signal from level k to a higher level l and back to level k again. Since the analysis operators $\psi^{\uparrow}_{k}$, $k \geq 0$, are designed to reduce signal information, they are not invertible in general. Therefore, $\hat{\psi}_{k,l}$ approximates a signal at level k by predicting it (by means of $\psi^{\downarrow}_{l,k}$) from the reduced information at level l incurred by $\psi^{\uparrow}_{k,l}$. We denote by V^l_k the range of the approximation operator, that is, $V^l_k = \mathrm{Ran}(\hat{\psi}_{k,l})$.

2.3.2 Pyramid condition

The analysis and synthesis operators $\psi^{\uparrow}_{k}$, $\psi^{\downarrow}_{k}$ are said to satisfy the *pyramid condition* if

$$\psi^{\uparrow}_{k}\psi^{\downarrow}_{k} = \mathrm{id} \quad \text{on} \quad V_{k+1}\,,$$

where id denotes the identity operator. This condition guarantees that the synthesis operator $\psi^{\downarrow}_{k}$ does not reduce information and thus, given a signal $\psi^{\downarrow}_{k}(x) \in V_k$, where $x \in V_{k+1}$, we are able to uniquely recover x. From here, it is easy to show that

$$\psi^{\uparrow}_{k,l}\psi^{\downarrow}_{l,k} = \mathrm{id} \quad \text{on} \quad V_l\,, \quad l > k\,.$$

In other words, the 'prediction' from $x \in V_l$ into a new signal $\psi^{\downarrow}_{l,k}(x)$ belonging to a finer level V_k is invertible. Moreover, it can be shown [63] that

$$V^l_k \subseteq V^{l-1}_k \subseteq V_k\,, \quad l > k+1\,, \tag{2.14}$$

where $V_k^l = \mathrm{Ran}(\hat{\psi}_{k,l})$ with

$$\hat{\psi}_{k,l} = \psi^{\downarrow}_{l,k}\psi^{\uparrow}_{k,l} = \psi^{\downarrow}_{k}\psi^{\downarrow}_{k+1}\cdots\psi^{\downarrow}_{k-1}\psi^{\uparrow}_{k-1}\psi^{\uparrow}_{k-2}\cdots\psi^{\uparrow}_{l}\,, \quad l > k\,.$$

In this case, the operator $\hat{\psi}_{k,l}$ maps the signal space V_k into nested subspaces $\cdots \subseteq V_k^{k+2} \subseteq V_k^{k+1} \subseteq V_k$, each subspace V_k^l containing all 'level l' ($l > k$) approximations of signals in V_k. Equation (2.14) is a basic requirement for multiresolution analyses [97] (see also Definition 1.4.1) that agrees with the intuition that the space V_k^{l-1}, which contains the approximations of signals at level k obtained by means of operator $\hat{\psi}_{k,l-1}$, contains the approximations of signals at level k obtained by means of $\hat{\psi}_{k,l}$ as well.

For the linear case where $\psi^{\uparrow}(x)(n) = (\bar{h} * x)(2n)$ and $\psi^{\downarrow}(x)(n) = (h * \check{x})(n)$ (see Example 2.3.1), it is easy to check that the pyramid condition amounts to

$$\sum_{l=-\infty}^{\infty} h(l)\bar{h}(2n-l) = \delta(n)\,. \tag{2.15}$$

Example 2.3.2 (Burt-Adelson pyramid with pyramid condition). We consider again the Burt and Adelson pyramid described in Example 2.3.1 but imposing the pyramid condition, which in this case reduces to (2.15). We assume that the analysis filter $\bar{h}$ has length 5 and we follow the design criteria proposed by Burt and Adelson [19] (see also (1.1)):

$$\bar{h}(0) = a\,, \quad \bar{h}(-1) = \bar{h}(1) = \frac{1}{4}\,, \quad \bar{h}(-2) = \bar{h}(2) = \frac{1}{4} - \frac{a}{2}\,, \quad \bar{h}(n) = 0 \quad \text{for other } n\,.$$

(a) Consider a symmetric synthesis filter of length 3:

$$h(0) = d\,, \qquad h(-1) = h(1) = e\,, \qquad h(n) = 0 \quad \text{for other } n.$$

The pyramid condition in (2.15) yields

$$\frac{1}{2}e + ad = 1 \qquad \text{and} \qquad (\frac{1}{4} - \frac{a}{2})d + \frac{1}{4}e = 0\,.$$

If we impose the normalizing condition $\sum_{l=-\infty}^{\infty} (-1)^l h(l) = 0$, we get a unique solution:

$$\bar{h}(-2) = \bar{h}(2) = -\frac{1}{8}\,, \quad \bar{h}(-1) = \bar{h}(1) = \frac{1}{4}\,, \quad \bar{h}(0) = \frac{3}{4}\,, \quad \bar{h}(n) = 0 \quad \text{for other } n,$$
$$h(-1) = h(1) = \frac{1}{2}\,, \quad h(0) = 1\,, \quad h(n) = 0 \quad \text{for other } n.$$

Or equivalently,

$$\psi^{\uparrow}(x)(n) = \tfrac{1}{8}\big(-x(2n-2) + 2x(2n-1) + 6x(2n) + 2x(2n+1) - x(2n+2)\big)$$
$$\begin{cases} \psi^{\downarrow}(x)(2n) = x(n) \\ \psi^{\downarrow}(x)(2n+1) = \tfrac{1}{2}\big(x(n) + x(n+1)\big)\,. \end{cases}$$

(*b*) Consider now a symmetric synthesis filter of length 5. The pyramid condition leads to

$$\bar{h}(-1) = \bar{h}(1) = \frac{1}{4}, \quad \bar{h}(0) = \frac{1}{2}, \quad \bar{h}(n) = 0 \quad \text{for other } n,$$
$$h(-2) = h(2) = -\frac{1}{4}, \quad h(-1) = h(1) = \frac{1}{2}, \quad h(0) = \frac{3}{2}, \quad h(n) = 0 \quad \text{for other } n.$$

Before we conclude this subsection, we point out the relationship between our framework and the frame reconstruction of the Laplacian pyramid proposed by Do and Vetterli in [49], where they study the Laplacian pyramid using frame theory. They show that the usual reconstruction (see (2.13) and synthesis part in Fig. 2.4) is suboptimal in the presence of noise and propose a new reconstruction method based on projection, which is the pseudo-inverse in certain cases. Do and Vetterli propose a reconstruction scheme (in the linear case) such as depicted in Fig. 2.5. Now, perfect reconstruction is achieved if and only if $\psi^{\downarrow}\psi^{\uparrow}$ is a projector or $\psi^{\uparrow}\psi^{\downarrow} = \text{id}$. Note that the latter is equivalent to the pyramid condition.

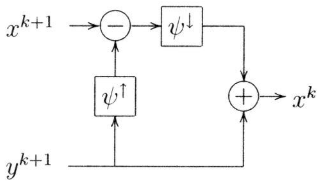


Figure 2.5: *Alternative synthesis scheme for the Laplacian pyramid.*

Now, instead of (2.11) we get

$$\Psi^{\downarrow}(x, y) = \psi^{\downarrow}(x) - \psi^{\downarrow}\psi^{\uparrow}(y) + y, \quad \text{for } x \in V_{k+1}\,,\ y \in W_{k+1}\,, \tag{2.16}$$

where $W_{k+1} \subseteq V_k$ and $\psi^{\downarrow}\colon V_{k+1} \to V_k$. Define

$$\omega^{\downarrow}(y) = y - \psi^{\downarrow}\psi^{\uparrow}(y)\,.$$

Now, the perfect reconstruction condition in (2.1) can be reformulated as

$$\psi^{\downarrow}\psi^{\uparrow}(x) + \omega^{\downarrow}\omega^{\uparrow}(x) = x, \quad \text{for } x \in V_k\,,$$

where $\omega^{\uparrow}(x) = x - \psi^{\downarrow}\psi^{\uparrow}(x)$.

2.3.3 Lifting pyramids

The lifting scheme, introduced in Section 2.2 for general decompositions, can be applied to pyramid decompositions. Both update and prediction lifting steps yield a modified pyramid scheme. From (2.8)-(2.10) and (2.11), we get that the updated analysis and synthesis operators are:

$$\begin{aligned}
\psi_u^{\uparrow}(x) &= \psi^{\uparrow}(x) + U\bigl(x - \psi^{\downarrow}\psi^{\uparrow}(x)\bigr)\,, && x \in V_0\\
\omega_u^{\uparrow}(x) &= \omega^{\uparrow}(x) = x - \psi^{\downarrow}\psi^{\uparrow}(x)\,, && x \in V_0\\
\Psi_u^{\downarrow}(x,y) &= \Psi^{\downarrow}\left(x - U(y), y\right) = \psi^{\downarrow}\bigl(x - U(y)\bigr) + y\,, && x \in V_1\,, y \in W_1 \subseteq V_0\,,
\end{aligned}$$

where $U\colon V_0 \longrightarrow V_1$ is the update operator. It is not difficult to show that the pyramid condition is preserved by an update lifting step, i.e., $(\psi_u^{\uparrow}, \psi^{\downarrow})$ satisfies the pyramid condition if $(\psi^{\uparrow}, \psi^{\downarrow})$ does so. Note, however, that the new decomposition structure does no longer satisfy the assumption in (2.11). Note also that if $\psi^{\downarrow}$ is linear, then $\Psi_u^{\downarrow}$ corresponds to (2.16) with $\psi^{\uparrow} = U$. In contrast, the pyramid condition will, in general, no longer be valid after a prediction lifting step. In this case, the modified analysis and synthesis operators are:

$$\begin{aligned}
\psi_p^{\uparrow}(x) &= \psi^{\uparrow}(x)\,, && x \in V_0\\
\omega_p^{\uparrow}(x) &= x - P\left(\psi^{\downarrow}\psi^{\uparrow}(x)\right)\,, && x \in V_0\\
\Psi_p^{\downarrow}(x,y) &= \Psi^{\downarrow}\bigl(x, y + P(x)\bigr) = \psi^{\downarrow}(x) + P(x) + y\,, && x \in V_1\,, y \in W_1 \subseteq V_0\,,
\end{aligned}$$

where $P\colon V_1 \longrightarrow W_1$ is the prediction operator. Note that in this case the assumption in (2.11) is satisfied with $\psi_p^{\downarrow} = \psi^{\downarrow} + P$ instead of $\psi^{\downarrow}$.

2.4 The wavelet transform

2.4.1 Axiomatics

A general wavelet decomposition has the structure depicted in Fig. 2.1, but in addition to the perfect reconstruction condition (2.1), i.e.,

$$\Psi^{\downarrow}\left(\psi^{\uparrow}(x), \omega^{\uparrow}(x)\right) = x, \quad \text{for } x \in V_0\,, \tag{2.17}$$

it satisfies the additional constraints

$$\begin{aligned}
\psi^{\uparrow}\left(\Psi^{\downarrow}(x,y)\right) &= x\,, \quad \text{for } x \in V_1\,, y \in W_1 && (2.18)\\
\omega^{\uparrow}\left(\Psi^{\downarrow}(x,y)\right) &= y\,, \quad \text{for } x \in V_1\,, y \in W_1\,, && (2.19)
\end{aligned}$$

which guarantee that the decomposition is non-redundant. Note that (2.17)-(2.19) imply that the analysis operator $\Psi^{\uparrow} = (\psi^{\uparrow}, \omega^{\uparrow})$ and the synthesis operator $\Psi^{\downarrow}$ are inverses of each other. Concatenation of a series of analysis steps yields a multiresolution decomposition called the *wavelet transform*.

Often, e.g. in the linear case, the synthesis operator $\Psi^{\downarrow}$ is of the special form

$$\Psi^{\downarrow}(x,y) = \psi^{\downarrow}(x) + \omega^{\downarrow}(y)\,, \quad x \in V_1\,, y \in W_1\,.$$

In this case, we speak of an *uncoupled wavelet decomposition* and conditions (2.17)-(2.19) become

$$\begin{aligned}
\psi^{\downarrow}\psi^{\uparrow}(x) + \omega^{\downarrow}\omega^{\uparrow}(x) &= x\,, \quad x \in V_0 && (2.20)\\
\psi^{\uparrow}\big(\psi^{\downarrow}(x) + \omega^{\downarrow}(y)\big) &= x\,, \quad x \in V_1\,, y \in W_1 && (2.21)\\
\omega^{\uparrow}\big(\psi^{\downarrow}(x) + \omega^{\downarrow}(y)\big) &= y\,, \quad x \in V_1\,, y \in W_1\,. && (2.22)
\end{aligned}$$

Fig. 2.6 depicts the corresponding synthesis part of an uncoupled wavelet decomposition for the multilevel case where $K=3$.

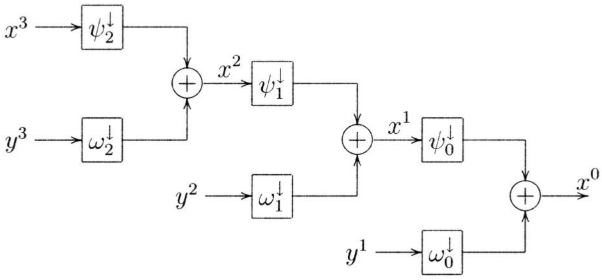


Figure 2.6: *Synthesis scheme of a 3-level uncoupled wavelet decomposition system.*

It was explained by Heijmans and Goutsias [70] how existing linear wavelets or filter banks (see Section 1.4) can fit into this abstract wavelet scheme. One can easily establish the relation between the biorthogonal conditions stated in (1.15)-(1.16) and (2.20)-(2.22). Note, however, that the expressions in (2.20)-(2.22) (and more general (2.17)-(2.19)) are formulated in operator terms, and do not require any sort of linearity assumption or inner product. This allows a broad class of nonlinear wavelet decomposition schemes [46, 55, 57, 66, 70].

2.4.2 Pyramid condition

It is straightforward to see that (in an uncoupled wavelet decomposition) the operators $\psi^{\uparrow}$, $\psi^{\downarrow}$ obey the pyramid condition, that is, $\psi^{\uparrow}\psi^{\downarrow} = \mathrm{id}$. In the linear case, where we have $\psi^{\uparrow}(x)(n) = (\bar{h}*x)(2n)$, $\psi^{\downarrow}(x)(n) = (h*\check{x})(n)$, the pyramid condition corresponds to one of the biorthogonal conditions, namely,

$$\sum_{l=-\infty}^{\infty} h(l)\bar{h}(2n-l) = \delta(n)\,.$$

In fact, it can be shown [70] that, in the linear case, any pyramid decomposition that satisfies the pyramid condition can be extended to a wavelet decomposition.

Example 2.4.1 (Burt-Adelson wavelet). Consider the Burt-Adelson pyramid discussed in Example 2.3.2(a). Then, a possible solution to biorthogonality conditions (1.15)-(1.16) is

$$\begin{array}{llll} \bar{g}(0) = \frac{1}{2}, & \bar{g}(1) = -1, & \bar{g}(2) = \frac{1}{2}, & \bar{g}(n) = 0 \quad \text{for other } n, \\ g(-3) = g(1) = \frac{1}{8}, & g(-2) = g(0) = \frac{1}{4}, & g(-1) = -\frac{3}{4}, & g(n) = 0 \quad \text{for other } n. \end{array}$$

This yields the following detail analysis and synthesis operators:

$$\omega^{\uparrow}(x)(n) = \tfrac{1}{2}\big(x(2n) - 2x(2n+1) + x(2n+2)\big)$$

$$\begin{cases} \omega^{\downarrow}(y)(2n) = \tfrac{1}{4}\big(y(n) + y(n+1)\big) \\ \omega^{\downarrow}(y)(2n+1) = \tfrac{1}{8}\big(y(n) - 6y(n+1) + y(n+2)\big) . \end{cases}$$

2.4.3 Lifting wavelets

The general technique of lifting described in Section 2.2 can also be applied for wavelet decompositions. In fact, the lifting scheme was introduced by Sweldens as a technique for building new wavelets from existing ones [146]. It is easy to show that the modified scheme resulting from lifting satisfies the constraints for a wavelet scheme, i.e., (2.17)-(2.19), and hereby also the pyramid condition. It is important to note, however, that lifting will in general give rise to coupled wavelet decompositions even if the original wavelet transform is uncoupled.

The lifting scheme has changed the wavelet scene dramatically since it provides a simple yet flexible method for the construction of new, possibly nonlinear, wavelets from existing ones. The initial wavelet transform can be a trivial one. 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., $x^1(n) = x^0(2n)$, $y^1(n) = x^0(2n+1)$. This latter decomposition, known in signal processing as polyphase decomposition, is sometimes called the *lazy wavelet transform*.

In the context of linear wavelets, it has been shown that any system using finite impulse response filters can be decomposed into elementary lifting steps [45]. Furthermore, as we observed before, lifting does not require that the prediction and the update operator are linear 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 the next chapter, we will exploit this fact by adapting the update operator to the local properties of the signal, hence we will design adaptive wavelet decompositions.

2.5 Other multiresolution decompositions

In the previous sections we have focused on pyramids and wavelets. However, there are many alternatives; for example, the wavelet packet representation [34], the local basis decompositions [99] or the multiwavelet construction [62, 145]. We briefly discuss some of these decompositions.

2.5.1 Wavelet packets

With a straightforward generalization of the wavelet transform, we can obtain an even sparser representation of a signal. Instead of decomposing only the approximation spaces V_k to derive the approximation and detail spaces V_{k+1} and W_{k+1}, we decompose the detail spaces as well. The recursive splitting of spaces can be represented in a binary tree as illustrated in Fig. 2.7. At each node of the tree, we have the option to split or not. This allows the construction of an arbitrary dyadic tree structure. Each structure is associated with a function basis known as a *wavelet packet basis* [34, 99]. One possibility is the octave-band tree, with low-pass iterations only; see Fig. 2.7(a). In fact, the standard wavelet basis is an example of a wavelet packet basis obtained by choosing such a tree. Here, the frequency axis is decomposed in dyadic intervals whose sizes have an exponential growth as shown in the time-frequency plane of Fig. 1.1. As discussed in Section 1.1.2, when the scale decreases, the frequency support of the wavelet is shifted toward high frequencies. The time resolution increases but the frequency resolution decreases. Thus, standard wavelet bases are suited for signals where high-frequency components have shorter duration than low-frequency components.

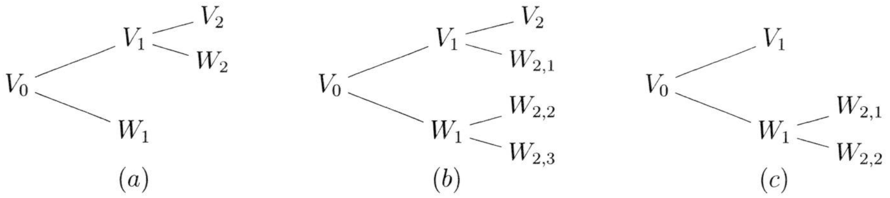


Figure 2.7: *Examples of tree-structure filter banks of depth 2: (a) only low-pass is split (standard wavelet); (b) full tree; (c) only high-pass is split.*

Wavelet packets generalize the fixed dyadic construction of the standard wavelet basis corresponding with Fig. 1.1 by decomposing the frequency axis in intervals of varying sizes, depending on the tree-structure filter bank which has been used. See Fig. 2.8 for an example.

Given a signal, we can obtain the 'best' (according to some criterion) tree decomposition. The best wavelet packet basis algorithm described in [168] is often used to select a 'best' basis that minimizes a concave cost function.

Wavelet packets are particularly well adapted to decompose signals that have different behavior in different frequency intervals. Note, however, that time-frequency tiles at a particular frequency have the same time resolution. Thus, if the signal is non-stationary in time, it is more appropriate to decompose the signal in a block basis that segments the time axis instead of the frequency axis. The local bases described next are examples of such a basis.

2.5.2 Local basis

A local basis divides the time axis into intervals of varying sizes, as shown in Fig. 2.9. Of particular interest are the cosine bases [33,99], which are obtained by designing smooth windows

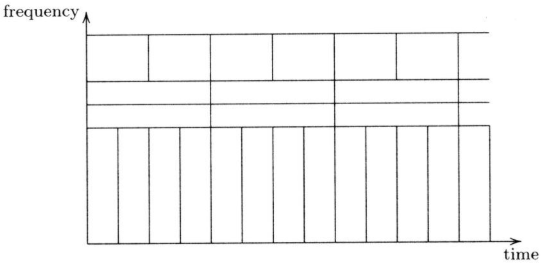


Figure 2.8: *A wavelet packet basis divides the frequency axis in separate intervals of varying sizes. A tiling is obtained by translating in time the wavelet packet basis covering each frequency interval.*

that cover each time interval and multiplying them by cosine functions of different frequencies.

Similarly to wavelet packets, a local cosine tree can be constructed by recursively dividing spaces built with local cosine bases. As in the wavelet packet case, this offers the possibility of choosing a 'best' basis for a given signal. A best local cosine basis adapts the time segmentation to the variations of the signal time-frequency structures. In comparison with wavelet packets, we gain time adaptation but we lose frequency flexibility (since the frequency axis is being split with constant bandwidth).

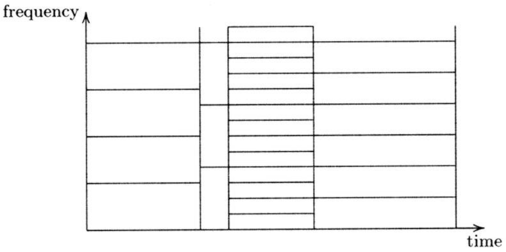


Figure 2.9: *A local basis divides the time axis in separate intervals of varying sizes. A tiling is obtained by translating in frequency the local basis covering each time interval.*

Note that wavelet packets and local cosines are dual families of bases. Wavelet packets segment the frequency axis and are uniformly translated in time, whereas local cosines divide the time axis and are uniformly translated in frequency. By combining the two dual concepts one can obtain arbitrary tilings of the time-frequency plane [72].

2.5.3 Multiwavelets

Multiwavelet decompositions offer more design flexibility by introducing at each level several analysis and synthesis operators. Multiwavelets have some advantages over scalar wavelets in relation to properties which are known to be important in signal processing such as short support, orthogonality, symmetry and vanishing moments. A scalar wavelet, except for the Haar case, cannot possess all these properties at the same time. In contrast, a multiwavelet system can simultaneously provide perfect reconstruction, orthogonality, linear-phase symmetry and a high number of vanishing moments [144]. The main drawback, however, is that they are implemented with more complicated filter banks than the standard wavelet transforms.

The standard wavelet transform is a scalar transform: it uses one analysis operator for the computation of the approximation signal, and one analysis operator for the computation of the detail signal. A multiwavelet transform generalizes the scalar wavelet transform: the approximation signal is generated by M analysis operators and the detail signal by M analysis operators.

In the same way that the scalar wavelet transform may be obtained by iterating a two-channel filter bank on its low-pass output (see Section 1.4.3), a multiwavelet transform can be obtained by a matrix-valued filter bank with 'coefficients' that are $M \times M$ matrices. Each input sample $\boldsymbol{x}(n)$ is a vector with M components The resulting two-channel $M \times M$-matrix filter bank operates on M input data streams, filtering them into $2M$ output streams.

Let us illustrate this for $M = 2$. We start with a signal x^0 which is pre-processed (e.g, by repeating each sample) to produce the sequences x_1^0 and x_2^0. Their coarse approximation is computed with the low-pass channel of the multiwavelet filter bank:

$$\begin{pmatrix} x_1^1(n) \\ x_2^1(n) \end{pmatrix} = \sum_{l=-\infty}^{\infty} \bar{\boldsymbol{h}}(l) \begin{pmatrix} x_1^0(2n-l) \\ x_2^0(2n-l) \end{pmatrix},$$

where each 'coefficient' $\bar{\boldsymbol{h}}(l)$ is a 2×2 matrix. Analogously, the details are computed with the high-pass channel of the multiwavelet filter bank:

$$\begin{pmatrix} y_1^1(n) \\ y_2^1(n) \end{pmatrix} = \sum_{l=-\infty}^{\infty} \bar{\boldsymbol{g}}(l) \begin{pmatrix} x_1^0(2n-l) \\ x_2^0(2n-l) \end{pmatrix}.$$

This one-level decomposition is shown in Fig. 2.10.

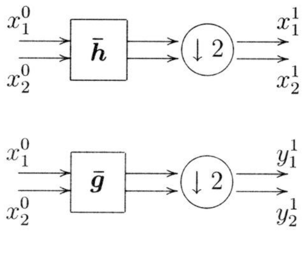


Figure 2.10: *Multiwavelet analysis filter bank with* $M = 2$. *An input* x^0 *is vectorized into* $\boldsymbol{x}^0 = (x_1^0, x_2^0)^T$. *Each row of the multifilter* $\bar{\boldsymbol{h}}$ *(and the same applies to* $\bar{\boldsymbol{g}}$*) is a combination of two scalar filters, one operating on* x_1^0 *and the other operating on* x_2^0.

Full multiwavelet decomposition of the signal x^0 is obtained by iterative filtering of the approximation coefficients $\boldsymbol{x}^k(n) = \left(x_1^k(n), x_2^k(n)\right)^T$ (where the superindex 'T' denotes transposition). The original signal can be reconstructed from the multiwavelet coefficients by means of the synthesis equation:

$$\boldsymbol{x}^k(n) = \sum_{l=-\infty}^{\infty} \boldsymbol{h}(l)\boldsymbol{x}^{k+1}(n-2l) + \sum_{l=-\infty}^{\infty} \boldsymbol{g}(l)\boldsymbol{y}^{k+1}(n-2l)\,.$$

Here, $\boldsymbol{x}^{k+1} = (x_1^{k+1}, x_2^{k+1})^T$, $\boldsymbol{y}^{k+1} = (y_1^{k+1}, y_2^{k+1})^T$ and $\boldsymbol{h}$, $\boldsymbol{g}$ are the synthesis multifilters. As in the scalar case, the filters need to satisfy the biorthogonality conditions:

$$\begin{aligned}
\sum_{l=-\infty}^{\infty} \boldsymbol{h}(l)\bar{\boldsymbol{h}}(2n-l) = \sum_{l=-\infty}^{\infty} \boldsymbol{g}(l)\bar{\boldsymbol{g}}(2n-l) &= \delta(n)\,\mathsf{I} \\
\sum_{l=-\infty}^{\infty} \boldsymbol{h}(l)\bar{\boldsymbol{g}}(2n-l) = \sum_{l=-\infty}^{\infty} \boldsymbol{g}(l)\bar{\boldsymbol{h}}(2n-l) &= \mathsf{O}\,,
\end{aligned}$$

where I, O denote the 2×2 identity and null matrices respectively.

Note that we have M times as many filters as in the classical wavelet case. One input signal x produces $2M$ outputs from the analysis bank. This means that the signal x has to be 'vectorized' so that M input streams $(x_1, \ldots, x_M)$ go together. The most obvious way to get M inputs from a given signal is to repeat the signal so that M identical streams go into the multifilter bank. This produces an overcomplete representation. A different way is to pre-process the given scalar signal so that if the data enters at rate r, pre-processing yields M streams at rate r/M for input to the multifilter, which then produces $2M$ output streams (M for the approximation and M for the detail), each at rate $r/2M$. In either case, the pre-processing should be reversible so that after the synthesis step, post-processing can recover the original signal x.

2.5.4 Steerable pyramid

The steerable pyramid is an overcomplete, linear, multiresolution and multiorientation image decomposition where the analysis and synthesis operators are (in the simplest case) derivative operators with different supports and orientations. The associated filters are such that the resulting transform is self-inverting (i.e., the synthesis filters are just a reflected version of the analysis filters) and, moreover, it is translation and rotation invariant. In the following, we briefly describe the construction and implementation of the steerable pyramid; a more detailed account can be found in [138].

The steerable pyramid transform is implemented as a filter bank consisting of polar-separable filters. For simplicity, we consider the filters in the frequency domain. Thus, a filter $h(m,n)$ in the spatial domain is expressed as $H(u,v)$ in the frequency domain, where u and v are the frequency variables corresponding to the two spatial directions. More precisely,

$$H(u,v) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h(m,n)e^{-ium}e^{-ivn}\,,$$

where $i = \sqrt{-1}$. For future convenience, we define $\boldsymbol{v} = (u, v)^T$ and $H^*(\boldsymbol{v}) = H(-\boldsymbol{v})$.

The frequency tiling after one level of decomposition is shown in Fig. 2.11 for the case of two orientation bands (i.e., $P = 2$). The corresponding diagram for this decomposition is depicted in Fig. 2.12. The filters B_p, $p = 1, 2$, are oriented band-pass filters, H_1 is a narrow-band low-pass filter, G_0 is a non-oriented high-pass filter and H_0 is a low-pass filter.

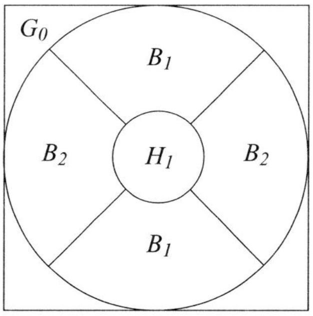


Figure 2.11: *Frequency tiling after 1-level decomposition of a steerable pyramid transform with $P = 2$. The frequency plane has been decomposed into a low-pass band (after filtering by H_1), two oriented band-pass components (after filtering by B_p, $p = 1, 2$), and a high-pass band (after filtering by G_0). The depicted squared region corresponds to the frequency range $[-\pi, \pi] \times [-\pi, \pi]$.*

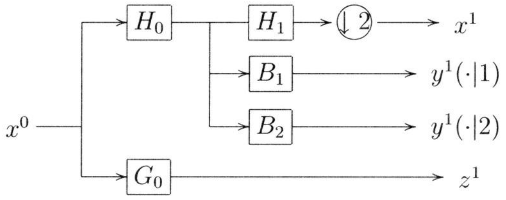


Figure 2.12: *First level of a steerable pyramid transform with $P = 2$.*

As illustrated in Fig. 2.11, the band-pass filters together act as a circular symmetric band-pass filter. The low-pass filter H_1 passes the low-frequency components that fall inside the central core of that circular filter, while the high-pass filter G_0 passes the high-frequency information that falls outside. In this way, the entire signal, regardless of its frequency, is passed to one of the output channels.

A block diagram illustrating the steerable pyramid transform and its inverse is shown in Fig. 2.13. Initially, the image is separated by the pre-processing filters H_0 and G_0 into a

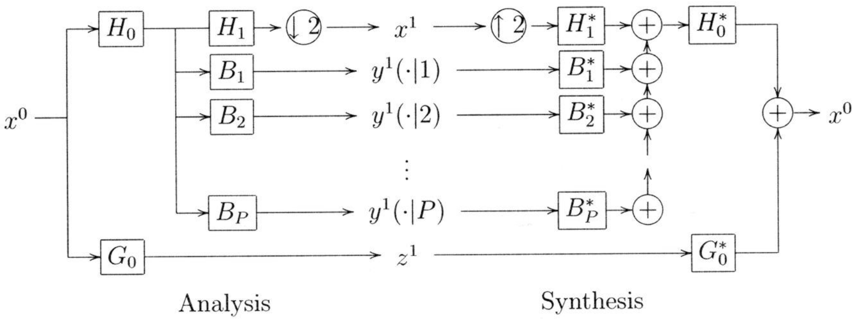


Figure 2.13: *Steerable pyramid transform (analysis) and its inverse (synthesis).*

low and a high-frequency bands. We denote the high-frequency band by z^1. The low-frequency band is then divided into a set of P oriented detail images and one approximation image. The detail images $y^1(\cdot|p)$, $p = 1, \dots, P$, are obtained using the band-pass filters $B_1, \dots, B_P$, while the approximation signal x^1 is obtained using a low-pass filter H_1 followed by a dyadic downsampling. The process of splitting into P details and one approximation is iterated on the approximation image (thus, filters H_0 and G_0 are not used in the successive levels).

In order to ensure that the transform is invertible as well as jointly invariant in orientation and space, the filters must satisfy specific radial (scale) and angular (orientation) frequency constraints [137]. The radial frequency constraints are:

1. Band limiting to prevent aliasing in the subsampling operation:

$$H_1(\boldsymbol{v}) = 0 \qquad \text{for } \|\boldsymbol{v}\| > \pi/2\,,$$

   where $\|\boldsymbol{v}\| = \sqrt{u^2 + v^2}$.

2. Flat system response to avoid amplitude distortion:

$$\left|H_0(\boldsymbol{v})\right|^2 \left(\left|H_1(\boldsymbol{v})\right|^2 + \sum_{p=1}^{P} \left|B_p(\boldsymbol{v})\right|^2\right) + \left|G_0(\boldsymbol{v})\right|^2 = 1\,. \tag{2.23}$$

3. Recursion. The low-pass channel of the system must be unaffected by the iteration process:

$$\left|H_1(\boldsymbol{v}/2)\right|^2 \left(\left|H_1(\boldsymbol{v})\right|^2 + \sum_{p=1}^{P} \left|B_p(\boldsymbol{v})\right|^2\right) = \left|H_1(\boldsymbol{v}/2)\right|^2\,. \tag{2.24}$$

A sufficient condition for (2.24) to hold is that the decomposition/reconstruction filter bank has unity gain for low frequencies:

$$\left|H_1(\boldsymbol{v})\right|^2 + \sum_{p=1}^{P} \left|B_p(\boldsymbol{v})\right|^2 = 1\,.$$

In this case, (2.23) implies that the pre and post-processing steps must also have a unity gain, that is,

$$|H_0(\boldsymbol{v})|^2 + |G_0(\boldsymbol{v})|^2 = 1 .$$

Typically, $H_0(\boldsymbol{v}) = H_1(\boldsymbol{v}/2)$, so that the initial low-pass 'shape' of the filter is the same as that used within the iteration. Thus, during the iteration $H_1(\boldsymbol{v}/2)$ plays the role of the initialization filter $H_0(\boldsymbol{v})$.

The angular constraint on the band-pass filters B_p requires these filters to form a steerable basis[1] [138]. In the simple case where the basis functions of the decomposition are directional derivative operators, the angular constraint can be expressed as

$$B_p(\boldsymbol{v}) = B(\boldsymbol{v})\big(-i\cos(\theta - \theta_p)\big)^{P-1} ,$$

where $i = \sqrt{-1}$, $\theta = \arctan(v/u)$, $\theta_p = \pi\frac{p-1}{P}$ for $p = 1, \ldots, P$, and

$$B(\boldsymbol{v}) = \left(\sum_{p=1}^{P} |B_p(\boldsymbol{v})|^2\right)^{1/2} .$$

The pyramid can be designed to produce any number of orientation bands P, resulting in an overcomplete transform with a redundancy factor of $4P/3$.

2.5.5 Gradient pyramid

A gradient pyramid [18] is obtained by applying a gradient operator to each level of the Gaussian pyramid $\{x^k\}$, $k = 0, \ldots, K$. Each image x^k is filtered by a set of four oriented gradient filters g_p, $p = 1, \ldots, 4$. The resulting filtered bands correspond to the detail images $y^{k+1}(\cdot|p)$, $p = 1, \ldots, 4$, representing the horizontal, vertical and the two diagonal directions. To reconstruct the original image from this gradient decomposition, a Laplacian pyramid is constructed as intermediate result. First, a (derivative) synthesis filter g_p is applied to $y^{k+1}(\cdot|p)$, $p = 1, \ldots, 4$. A Laplacian pyramid $\{y_L^{k+1}\}$, can then be obtained by summing up, at each level, the filtered resulting images. Fig. 2.14 illustrates one level of the gradient pyramid transform. Here, $\bar{h}$ is the low-pass filter used to construct the Gaussian pyramid $\{x^k\}$, $k = 0, \ldots, K$, and h its corresponding synthesis filter. In [18], the filter $\bar{h}$ is assumed to be of the form $\bar{h} = \dot{h} * \dot{h}$ with $\dot{h}$ being a 3×3 binomial filter. Then, it can be shown that perfect reconstruction is possible by taking

$$\bar{g}_p = (1 + \dot{h}) * d_p \qquad \text{and} \qquad g_p = -\frac{1}{8} d_p ,$$

where d_p, $p = 1, \ldots, 4$, are the derivative filters:

$$d_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad d_2 = \begin{pmatrix} 1 & -1 \end{pmatrix}, \quad d_3 = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad d_4 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} .$$

---

[1] A set of filters forms a steerable basis if they are rotated versions of each other and a version of the filters at any orientation may be synthesized as a linear combination of the basis filters. The simplest example of a steerable basis is a set of P directional derivatives of order $P - 1$.

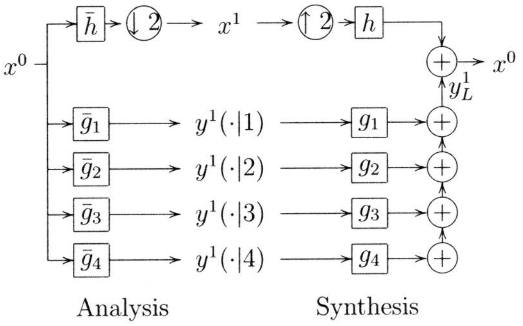


Figure 2.14: *Gradient pyramid transform (analysis) and its inverse (synthesis).*

Chapter 3

Adaptive update lifting: the axiomatic framework

Multiresolution (MR) representations, such as pyramids and wavelets, provide a powerful tool for the analysis of signals, images, and video sequences. Classical MR transforms lead to an isotropic smoothing of the signal when going to lower resolutions. However, for various applications there is a strong demand for a more 'high-level' analysis and thereby for MR representations that take into account the characteristics of the underlying signal and do leave intact or even enhance certain important signal characteristics such as sharp transitions, edges, singularities, local extrema and other geometric structures of interest. The importance of such 'adaptive' or 'data-driven' representations has led to a wealth of new directions in MR approaches such as bandelets, ridgelets, curvelets, morphological wavelets, etc., which go beyond standard wavelet theory. These systems combine ideas of multiresolution analysis with notions of geometric features and structures in order to build decompositions which are suited to a given task. Often, this can be achieved by making the decomposition dependent on the underlying data.

In this chapter, we propose a technique for building adaptive wavelets by means of an adaptive update lifting step followed by a fixed prediction lifting step. The adaptivity consists hereof that the system can choose between different update filters, and that this choice depends on the information locally available within the input bands (e.g., the gradient of the original signal). In this way, only homogeneous regions are smoothed while discontinuities are preserved.

The chapter is organized as follows. Section 3.1 introduces the idea of adaptive wavelets and recalls some of the existing approaches in the literature. Section 3.2 presents a general framework for building non-redundant adaptive wavelets by update lifting. The rest of the chapter deals with a special class of adaptive update lifting scheme where the system can choose between two different update steps depending on the local gradient of the signal. After giving some mathematical notions in Section 3.3, the update filters and the decision function which triggers the choice of these filters are discussed in more detail in Section 3.4. Necessary and sufficient conditions for perfect reconstruction of such an adaptive system are derived in Section 3.5.

3.1 Adaptive wavelets: existing approaches

Originally, wavelet transforms were linear, and their construction was based on classical tools from functional and harmonic analysis such as the Fourier transform. However, classical wavelet transforms are not always suitable to analyze discontinuities encountered in real-world signals, in the sense that they perform a uniform smoothing which does not take into account the geometric structure of the signal. Moreover, such discontinuities (e.g., sharp transitions in one-dimensional signals and edges in images) tend to give rise to large coefficients in their proximity, which is very undesirable in coding and compression applications. This has motivated a growing interest in finding new representations able to preserve important singularities in the signal while providing a compact representation.

Indeed, for most tasks in signal and image processing, such as texture analysis, segmentation, compression, denoising and deconvolution, it is of paramount importance that the representation at hand takes into account the geometric nature of the underlying signal. In other words, MR representations must *adapt* themselves to the signal structure. This can mean, for example, that the filters constituting a wavelet decomposition are 'shaped' or 'steered' by the input data. One can find several approaches to introduce some kind of adaptivity into an MR decomposition. In what follows we discuss some of these approaches.

A first approach to adaptivity is to use arbitrary subband decomposition trees (wavelets packets or local cosine bases) to choose a basis depending on the signal. The *best basis algorithm* [168], for example, selects a wavelet basis by minimizing a concave cost function such as the entropy or an l^p-norm. To further characterize the space-varying characteristics, the spatially adaptive wavelet packets were introduced in [72,118] by performing a spatial segmentation and adapting the wavelet packet frequency decomposition to each spatial segment. Similarly, adaptive local cosine basis decompositions [32] as well as jointly adaptive space and frequency basis decompositions [73] have been proposed for a better space-frequency representation. In such approaches, the filter coefficients are fixed for an entire block of data as the optimization criterion is a global one. However, using a single prototype filter may not characterize well the local variations of the signal.

Another approach is to look for bases that are capable of 'tracking' the shape of the discontinuities. This has led to construction of functions whose support has a shape that can be adapted to the regularity of the signal being analyzed. Donoho [50] studies the optimal approximations of particular classes of signals with an overcomplete collection of elementary functions called *wedgelets*. His construction is based on a multiscale organization of the edge data. Another construction due to Donoho [51] are the *ridgelets*. These are elongated functions especially suited for object discontinuities across straight lines. Motivated by the problem of finding efficient representations of objects with discontinuities along curves, Candès and Donoho [23] introduced yet another representation system, the *curvelet* transform. Curvelets are based on multiscale ridgelets combined with a spatial band-pass filtering operation to isolate different scales. It has been shown that, under certain assumptions, curvelet frames have optimal approximation properties for two-dimensional functions which are piecewise constant. This has led to different constructions and applications for the curvelet transform [24,142].

Recently, *bandelets* have been proposed by Le Pennec and Mallat [88]. They construct 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 wavelet bases. The resulting wavelets have their support in a band surrounding the edge curve, hence the name bandelet.

Another MR representation for images which incorporate a specific geometric treatment of edges is proposed by Cohen and Matei [31]. Their approach is based on the nonlinear MR representation of Harten [67] while incorporating edge detection within the same transform.

Chan and Zhou [26] extend the essentially non-oscillatory (ENO) technique[1] [68] to modify the standard wavelet transform near discontinuities. 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.

The introduction of the lifting scheme by Sweldens [146–148] opened the way to the design of nonlinear wavelet transforms [46, 55, 57, 65, 66, 70]. In all these approaches, the flexibility and freedom offered by the lifting scheme were 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 always with the sudden changes in the input signal. To overcome this problem, various lifting schemes with space-varying prediction filters have been proposed.

Trappe and Liu [160] build adaptivity into the prediction step of the lifting scheme. Their aim is to design a data-dependent prediction filter to minimize 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 minimized using Wiener filter theory. Their second approach is based on the classical adaptive filter theory for designing time-varying filter banks [69]. It uses a local optimization criterion and, in this case, the coefficients of the prediction filter vary over time (or space). Here the filter coefficients at a given location n are updated using the approximation signal x and the predicted detail y' at location $n - 1$. In this scheme, perfect reconstruction is automatic. A similar approach had been earlier proposed by Gerek and Çetin [60, 61]. These latter approaches are *causal* in the sense that the computation of the detail signal at a given location depends 'only' on *previously* computed detail samples. That is, the detail sample $y(n)$ is not used for determining the prediction filter at location n. This differs from our scheme, to be introduced in the next section, where both $x(n)$ and $y(n)$ are used for the computation of the filter coefficients.

Claypoole *et al.* [29, 30] propose an adaptive lifting scheme, which they call *space-adaptive transform*, which lowers the order of the approximation near jumps to avoid prediction across discontinuities. In [30], the choice of prediction filter depends only on the approximation signal and thus, this approach still fits within the classical lifting framework (where perfect reconstruction is guaranteed), albeit that the lifting operator is nonlinear in this case. The approach presented in [29], however, does not fit within the classical lifting scheme as the prediction step does require input from both channels. To guarantee perfect reconstruction at

---

[1] The basic idea behind an ENO scheme is to construct a piecewise polynomial approximation of a given function by using only information from smooth regions.

synthesis, one has to keep track of the filter choices made at each sample. As a consequence, the resulting decomposition is no longer non-redundant.

Our approach resembles the approach by Claypoole *et al.* [29] in the sense that it does not fit in the classical lifting scheme either. However, we choose 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. Therefore, an important feature of our adaptive representation is that it is *neither causal nor redundant.*

3.2 General framework for update lifting

Assume that an input signal $x^0\colon \mathbb{Z}^d \to \mathbb{R}$, henceforth denoted by x_0, is decomposed into two components x and y, where possibly y comprises more than one band, i.e.,

$$y = \{y(\cdot|1), \ldots, y(\cdot|P)\} \qquad \text{with } P \geq 1\,. \tag{3.1}$$

The bands $x, y(\cdot|1), \ldots, y(\cdot|P)$, which generally represent the polyphase components of the analyzed signal x_0, are the input bands for our lifting scheme. In any case, we assume that the decomposition $x_0 \mapsto (x, y)$ is invertible and hence we can perfectly reconstruct x_0 from its components x and y. The first signal x will be updated in order to obtain an approximation signal x' whereas $y(\cdot|1), \ldots, y(\cdot|P)$ will be further predicted so as to generate a detail signal $y' = \{y'(\cdot|1), \ldots, y'(\cdot|P)\}$. In our lifting scheme, the update step is adaptive while the prediction step is fixed. This implies that the signal y can be easily recovered from the approximation x' and the detail y'. The recovery of x from x' and y is less obvious. Henceforth, we concentrate on the update lifting step.

The basic idea underlying our adaptive scheme is that the update parameters depend on the information locally available within both signals x and y, as shown in Fig. 3.1. In this scheme D is a *decision map* which uses inputs from all bands, i.e., $D = D(x, y) = D\big(x, y(\cdot|1), \ldots, y(\cdot|P)\big)$, and whose output is a decision parameter d which governs the choice of the update step: for every possible decision d of the decision map, we have a different update operator U_d and addition $\oplus_d$. More precisely, if $d_{\boldsymbol{n}}$ is the output of D at location $\boldsymbol{n} \in \mathbb{Z}^d$, then the updated value $x'(\boldsymbol{n})$ is given by

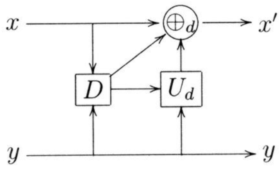


Figure 3.1: *Adaptive update lifting scheme.*

$$x'(\boldsymbol{n}) = x(\boldsymbol{n}) \oplus_{d_{\boldsymbol{n}}} U_{d_{\boldsymbol{n}}}(y)(\boldsymbol{n}) = x(\boldsymbol{n}) \oplus_{d_{\boldsymbol{n}}} U_{d_{\boldsymbol{n}}}\big(y(\cdot|1), \ldots, y(\cdot|P)\big)(\boldsymbol{n})\,, \tag{3.2}$$

and can be inverted by means of

$$x(\boldsymbol{n}) = x'(\boldsymbol{n}) \ominus_{d_n} U_{d_n}(y)(\boldsymbol{n}) = x'(\boldsymbol{n}) \ominus_{d_n} U_{d_n}\big(y(\cdot|1), \dots, y(\cdot|P)\big)(\boldsymbol{n}), \tag{3.3}$$

where $\ominus_d$ denotes the subtraction which inverts $\oplus_d$. Thus, presumed that d is known for every location $\boldsymbol{n}$, we can recover the original signal x, and hence have perfect reconstruction.

The invertibility of such a scheme is far from trivial if we want to avoid the overhead of storing the decision map (i.e., the decision parameter $d_{\boldsymbol{n}}$ for every $\boldsymbol{n}$). The reason is that $d_{\boldsymbol{n}} = D(x, y)(\boldsymbol{n})$ depends on the original signal x while at synthesis we do not know x but 'only' its update x'. In general, this prohibits the computation of $d_{\boldsymbol{n}}$ and in such cases perfect reconstruction is out of reach. However, as we will show later, under some special circumstances it is possible to recover $d_{\boldsymbol{n}}$ from x' and $y = \{y(\cdot|1), \dots, y(\cdot|P)\}$ by means of a so-called *posterior decision map* D'. Obviously, this map needs to satisfy

$$D'(x', y) = D(x, y),$$

for all inputs $x, y = \{y(\cdot|1), \dots, y(\cdot|P)\}$, with x' given by (3.2). 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 assume that at each location $\boldsymbol{n} \in \mathbb{Z}^d$ the update step depends only on $x(\boldsymbol{n})$ and N samples from signal y, say $y(\boldsymbol{n} + \boldsymbol{l}_j|p_j)$, where $\boldsymbol{l}_j \in \mathbb{Z}^d$ and $p_j \in \{1, \dots, P\}$ for $j = 1, \dots, N$. We use the notation:

$$y_j(\boldsymbol{n}) = y(\boldsymbol{n} + \boldsymbol{l}_j|p_j), \quad j = 1, \dots, N.$$

Note that we have some freedom in labeling the values $y(\boldsymbol{n}+\boldsymbol{l}|p)$ by j. Fortunately, the specific choice of the labeling is of no importance. We give two examples to illustrate this notation.

Example 3.2.1. First we consider the one-dimensional case with only two input bands x and y (hence P=1). Assume that samples $x_0(2n), x_0(2n+1)$ of the original signal correspond with samples $x(n), y(n)$, respectively, and that $x(n)$ is updated with its two neighbors $y(n-1)$ and $y(n)$. Thus, $N = 2$ and we could, for example, label $y_1(n) = y(n-1)$ and $y_2(n) = y(n)$ as shown in Fig. 3.2. Obviously another choice is $y_1(n) = y(n)$ and $y_2(n) = y(n-1)$. Note that in both cases the labelings are not one-to-one: for example, for the former choice, $y_2(n) = y_1(n+1)$.

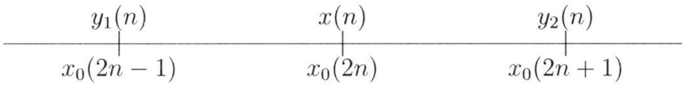


Figure 3.2: *Example of indexing the input samples for one-dimensional signals.*

Example 3.2.2. Next, we consider two-dimensional signals as depicted in Fig. 3.3. Here, we assume a decomposition with $P = 3$ corresponding with a square (i.e., 2×2) sampling structure. The geometrical interpretation of the three last band signals is as follows (see also

$(2m-1, 2n-1)$	$(2m-1, 2n)$	$(2m-1, 2n+1)$
$(2m, 2n-1)$	$(2m, 2n)$	$(2m, 2n+1)$
$(2m+1, 2n-1)$	$(2m+1, 2n)$	$(2m+1, 2n+1)$

$y(\boldsymbol{n}-\boldsymbol{a}-\boldsymbol{b}\|3)$	$y(\boldsymbol{n}-\boldsymbol{a}\|2)$	$y(\boldsymbol{n}-\boldsymbol{a}\|3)$
$y(\boldsymbol{n}-\boldsymbol{b}\|1)$	$x(\boldsymbol{n})$	$y(\boldsymbol{n}\|1)$
$y(\boldsymbol{n}-\boldsymbol{b}\|3)$	$y(\boldsymbol{n}\|2)$	$y(\boldsymbol{n}\|3)$

Figure 3.3: *Left: coordinates for two-dimensional signals. Right: location of the input signals* x *and* $y(\cdot|1), y(\cdot|2), y(\cdot|3)$ *after square sampling. Here* $\boldsymbol{a} = (1,0)^T$ *and* $\boldsymbol{b} = (0,1)^T$.

the right diagram in Fig. 3.3): after prediction, $y(\cdot|1), y(\cdot|2)$ will represent the detail bands capturing *vertical* and *horizontal* details, respectively. The interpretation of $y(\cdot|3)$ is somewhat less intuitive. After prediction, it leads to what is usually called the *diagonal* detail band.

Let us assume that $x(\boldsymbol{n})$ is updated with its eight horizontal, vertical and diagonal neighbors. This involves the samples (starting at the east and rotating counter-clockwise): $y(\boldsymbol{n}|1), y(\boldsymbol{n} - \boldsymbol{a}|3), y(\boldsymbol{n}-\boldsymbol{a}|2), y(\boldsymbol{n}-\boldsymbol{a}-\boldsymbol{b}|3), y(\boldsymbol{n}-\boldsymbol{b}|1), y(\boldsymbol{n}-\boldsymbol{b}|3), y(\boldsymbol{n}|2)$ and $y(\boldsymbol{n}|3)$. Here $\boldsymbol{a}, \boldsymbol{b}$ are the unit row and column vectors $(1,0)^T$ and $(0,1)^T$, where the superindex 'T' denotes transposition. In this example, choosing a counter-clockwise labeling direction, we get $y_1(\boldsymbol{n}) = y(\boldsymbol{n}|1)$, $y_2(\boldsymbol{n}) = y(\boldsymbol{n} - \boldsymbol{a}|3)$, $y_3(\boldsymbol{n}) = y(\boldsymbol{n} - \boldsymbol{a}|2)$, etc., as depicted in Fig. 3.4. Again, this labeling is not one-to-one: e.g., $y_2(\boldsymbol{n}) = y_8(\boldsymbol{n} - \boldsymbol{a})$.

3.3 Intermezzo: seminorms

Before we give an explicit expression for the update step and examine the question under which assumptions it is invertible, we need to introduce the concept of seminorm and other notions that we will need in the sequel.

Definition 3.3.1. Let V be a vector space over $\mathbb{R}$. A function p: $V \longrightarrow \mathbb{R}_+$ is called a *seminorm* if the following two properties hold:

(*i*) $p(\lambda \boldsymbol{v}) = |\lambda|\, p(\boldsymbol{v}), \quad \boldsymbol{v} \in V,\ \lambda \in \mathbb{R}$

(*ii*) $p(\boldsymbol{v}_1 + \boldsymbol{v}_2) \le p(\boldsymbol{v}_1) + p(\boldsymbol{v}_2), \quad \boldsymbol{v}_1, \boldsymbol{v}_2 \in V$.

This last inequality is called the *triangle inequality*.

$y_4(\boldsymbol{n})$	$y_3(\boldsymbol{n})$	$y_2(\boldsymbol{n})$
$y_5(\boldsymbol{n})$	$x(\boldsymbol{n})$	$y_1(\boldsymbol{n})$
$y_6(\boldsymbol{n})$	$y_7(\boldsymbol{n})$	$y_8(\boldsymbol{n})$

Figure 3.4: *Example of indexing the input samples for two-dimensional signals.*

A large class of seminorms on $\mathbb{R}^N$ is given by the expression

$$p(\boldsymbol{v}) = \big(\sum_{i=1}^{I} |\boldsymbol{a}_i^T \boldsymbol{v}|^q\big)^{1/q}, \tag{3.4}$$

where $\boldsymbol{a}_i \in \mathbb{R}^N$, $i = 1, \ldots, I$, and $q \geq 1$. By $\boldsymbol{a}^T\boldsymbol{v}$ we mean the inner product of the vectors $\boldsymbol{a}$ and $\boldsymbol{v}$.

For example, if $q = 1$ and $I = 1$, we get

$$p(\boldsymbol{v}) = \left|\boldsymbol{a}^T\boldsymbol{v}\right| ,$$

which we simply refer to as *weighted vector seminorm*. The seminorms given by

$$p(\boldsymbol{v}) = \left(\boldsymbol{v}^T M \boldsymbol{v}\right)^{1/2} , \tag{3.5}$$

where M is a symmetric positive semi-definite matrix, are called *quadratic seminorms*. It is not difficult to show that they belong to the family given by (3.4) with $q = 2$. Indeed, if M is a symmetric positive semi-definite matrix, we can write [76]:

$$M = \sum_{i=1}^{N} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^T, \ \lambda_i \geq 0 ,$$

where $\{\lambda_i \mid 1 \leq i \leq N\}$ are the eigenvalues of M and $\{\boldsymbol{u}_i \mid 1 \leq i \leq N\}$ are the (orthogonal) eigenvectors of M. The expression (3.5) becomes

$$p(\boldsymbol{v}) = \left|\boldsymbol{v}^T\big(\sum_{i=1}^{N} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^T\big)\boldsymbol{v}\right|^{1/2} = \left(\sum_{i=1}^{N} \lambda_i |\boldsymbol{u}_i^T \boldsymbol{v}|^2\right)^{1/2} .$$

Now, if we take $\boldsymbol{a}_i = \sqrt{\lambda_i}\boldsymbol{u}_i$, we get (3.4) with $q = 2$ and $I = N$.

Recall that p is a norm if, in addition to (i)-(ii) in Definition 3.3.1, it satisfies $p(\boldsymbol{v}) = 0$ if and only if $\boldsymbol{v} = \boldsymbol{0}$. Obviously, every norm is a seminorm but not vice versa. In particular, the seminorm given in (3.5) is a norm when M is a symmetric positive definite matrix. A special case is the l^2-norm which results when M is the identity matrix. The well-known l^q-norms $(1 \le q < \infty)$ are obtained from (3.4) with $I = N$ and $\{\boldsymbol{a}_i \mid 1 \le i \le N\}$ being the canonical basis[2] of $\mathbb{R}^N$.

Let V be a vector space with seminorm p. For a linear operator $A\colon V \to V$ we define the *operator seminorm* $p(A)$ and the *inverse operator seminorm* $p^{-1}(A)$ as

$$\begin{aligned} p(A) &= \sup\{p(A\boldsymbol{v}) \mid \boldsymbol{v} \in V \text{ and } p(\boldsymbol{v}) = 1\} \\ p^{-1}(A) &= \sup\{p(\boldsymbol{v}) \mid \boldsymbol{v} \in V \text{ and } p(A\boldsymbol{v}) = 1\}\,. \end{aligned}$$

In the last expression we use the convention that $p^{-1}(A) = \infty$ if $p(A\boldsymbol{v}) = 0$ for all $\boldsymbol{v} \in V$, unless p is identically zero, in which case both $p(A)$ and $p^{-1}(A)$ are zero. Throughout the remainder, we will discard the case where p is identically zero and, consequently, we will always have $p^{-1}(A) > 0$.

We list some properties of these two notions in the following proposition.

Proposition 3.3.2. *Let V be Hilbert space, let $p\colon V \to \mathbb{R}_+$ be a seminorm and $A\colon V \to V$ be a bounded linear operator.*

- *(a) $p^{-1}(A) = p(A^{-1})$ if A is invertible.*
- *(b) The following two conditions are equivalent*
  - *(i) $p(A) < \infty$*
  - *(ii) $p(\boldsymbol{v}) = 0$ implies $p(A\boldsymbol{v}) = 0$ for $\boldsymbol{v} \in V$.*
- *(c) The following two conditions are also equivalent*
  - *(i) $p^{-1}(A) < \infty$*
  - *(ii) $p(A\boldsymbol{v}) = 0$ implies $p(\boldsymbol{v}) = 0$ for $\boldsymbol{v} \in V$.*
- *(d) $p(A\boldsymbol{v}) \le p(A)p(\boldsymbol{v})$ if $p(\boldsymbol{v}) \ne 0$.*
- *(e) $p(\boldsymbol{v}) \le p^{-1}(A)p(A\boldsymbol{v})$ if $p(A\boldsymbol{v}) \ne 0$.*

Proof. The proofs of (a), (d) and (e) are straightforward. We prove (b) and (c).

(b): Assume (i), that is $p(A) < \infty$. Now suppose that there exists a $\boldsymbol{v} \in V$ such that $p(\boldsymbol{v}) = 0$ and $p(A\boldsymbol{v}) \ne 0$. We show that this gives rise to a contradiction. Fix a vector $\boldsymbol{w} \in V$ with $p(\boldsymbol{w}) = 1$. If $\lambda \in \mathbb{R}$, then

$$p(\lambda\boldsymbol{v} + \boldsymbol{w}) \le |\lambda| p(\boldsymbol{v}) + p(\boldsymbol{w}) = 1\,,$$

---

[2] The vectors $\boldsymbol{a}_i \in \mathbb{R}^N$, $i = 1, \ldots, N$, are said to be a canonical basis of $\mathbb{R}^N$ if $\boldsymbol{a}_1 = (1, 0, \ldots, 0)^T, \boldsymbol{a}_2 = (0, 1, 0, \ldots, 0)^T, \ldots, \boldsymbol{a}_N = (0, \ldots, 0, 1)^T$.

and also

$$1 = p(\boldsymbol{w}) \leq p(\lambda\boldsymbol{v} + \boldsymbol{w}) + p(-\lambda\boldsymbol{v}) = p(\lambda\boldsymbol{v} + \boldsymbol{w}),$$

which means that

$$p(\lambda\boldsymbol{v} + \boldsymbol{w}) = 1 \text{ for every } \lambda \in \mathbb{R}.$$

By definition,

$$p(A) \geq p\big(A(\lambda\boldsymbol{v} + \boldsymbol{w})\big) \geq p(\lambda A\boldsymbol{v}) - p(A\boldsymbol{w}) = |\lambda| p(A\boldsymbol{v}) - p(A\boldsymbol{w}).$$

Letting $|\lambda| \to \infty$, we arrive at the conclusion that $p(A) = \infty$, a contradiction.

Assume now that *(ii)* holds. Define $V_0 \subseteq V$ as $V_0 = \{\boldsymbol{v} \in V \mid p(\boldsymbol{v}) = 0\}$ and $V_1 = V_0^{\perp}$. It is easy to see that for any $\boldsymbol{v} \in V$ we have $p(\boldsymbol{v}) = p(\boldsymbol{v}_1)$ where $\boldsymbol{v}_1$ is the projection of $\boldsymbol{v}$ on V_1. Obviously, p defines a norm on the closed subspace V_1. The decomposition of V into V_0 and V_1 gives rise to a decomposition of the operator A into A_{ij} where A_{ij} maps V_j into V_i, for $i, j = 0, 1$. Thus we can write

$$A\boldsymbol{v} = (A_{00}\boldsymbol{v}_0 + A_{01}\boldsymbol{v}_1) + (A_{10}\boldsymbol{v}_0 + A_{11}\boldsymbol{v}_1),$$

where the first and second expression between brackets lies in V_0 and V_1, respectively. The condition in *(ii)* obviously means that $A_{10} = 0$. It is then evident that

$$p(A) = \sup\{p(A_{11}\boldsymbol{v}_1) \mid p(\boldsymbol{v}_1) = 1\},$$

and this coincides with the norm of A_{11} on V_1 which, by definition, is finite. This proves *(b)*.

(c): This proof is very similar to that of *(b)*. In the second part of the proof where it has to be shown that $p^{-1}(A) < \infty$, it is found that $A_{10} = 0$, A_{11} is invertible, and $p^{-1}(A) = p(A_{11}^{-1})$, which is finite. □

3.4 Choice of decision map and update filters

We return to the framework of Section 3.2, and define the *gradient vector* $\boldsymbol{v}(\boldsymbol{n}) = \big(v_1(\boldsymbol{n}), \ldots, v_N(\boldsymbol{n})\big)^T \in \mathbb{R}^N$ by

$$v_j(\boldsymbol{n}) = x(\boldsymbol{n}) - y_j(\boldsymbol{n}), \quad j = 1, \ldots, N. \tag{3.6}$$

Recall that $y_j(\boldsymbol{n}) = y(\boldsymbol{n} + \boldsymbol{l}_j | p_j)$, where the samples $y(\boldsymbol{n} + \boldsymbol{l}_j | p_j)$ are those used by the update step.

We assume that the decision map at location $\boldsymbol{n}$ depends exclusively on the gradient vector $\boldsymbol{v}(\boldsymbol{n})$. Furthermore, in the remainder of this chapter we consider *binary* decision maps where d can only take the value 0 or 1, governed by a simple threshold criterion: if the gradient is large (in some seminorm sense) it chooses one filter, if it is small the other. In particular, we consider binary decision maps of the form:

$$D(x, y)(\boldsymbol{n}) = \begin{cases} 1, & \text{if } p\big(\boldsymbol{v}(\boldsymbol{n})\big) > T \\ 0, & \text{if } p\big(\boldsymbol{v}(\boldsymbol{n})\big) \leq T, \end{cases} \tag{3.7}$$

where $\boldsymbol{v}(\boldsymbol{n}) \in \mathbb{R}^N$ is the gradient vector given by (3.6), p: $\mathbb{R}^N \to \mathbb{R}_+$ is a seminorm, and $T > 0$ is a given threshold. Instead of (3.7) we may also use the shorthand notation

$$D(x, y)(\boldsymbol{n}) = [\,p\big(\boldsymbol{v}(\boldsymbol{n})\big) > T\,], \tag{3.8}$$

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

Not every seminorm can be used to model an adaptive scheme. For example, if $p\big(\boldsymbol{v}(\boldsymbol{n})\big)$ depends only on differences $v_i(\boldsymbol{n}) - v_j(\boldsymbol{n})$, then the decision criterion in (3.7) is independent of the value of $x(\boldsymbol{n})$, as can easily be seen by using (3.6). A simple condition on p which is necessary for the scheme to be truly adaptive is

$$p(\boldsymbol{u}) > 0\,,$$

where $\boldsymbol{u} = (1, \dots, 1)^T$ is a vector of length N. Indeed, it is easy to check that the condition $p(\boldsymbol{u}) = 0$ is equivalent to the condition

$$p(\boldsymbol{v} + \lambda \boldsymbol{u}) = p(\boldsymbol{v}),\;\; \boldsymbol{v} \in \mathbb{R}^N,\;\; \lambda \in \mathbb{R}\,.$$

Observe that the addition of λ to $x(\boldsymbol{n})$, while keeping all $y_j(\boldsymbol{n})$ constant, amounts to the addition of $\lambda \boldsymbol{u}$ to the gradient vector $\boldsymbol{v}(\boldsymbol{n})$. If such an addition does not affect the seminorm, then the corresponding decision criterion does not depend on $x(\boldsymbol{n})$, and hence the scheme is non-adaptive.

Adaptivity Condition for the Seminorm. The seminorm p on $\mathbb{R}^N$ satisfies

$$p(\boldsymbol{u}) > 0\,, \tag{3.9}$$

where $\boldsymbol{u} = (1, \dots, 1)^T$ is a vector of length N.

In the update step given by (3.2) we need to specify the 'addition' $\oplus_d$ as well as the update filter $U_d(y)(\boldsymbol{n})$, for the values $d = 0, 1$. Henceforth, we assume that the addition $\oplus_d$ is of the form:

$$x \oplus_d u = \alpha_d (x + u)\,, \qquad \text{with } \alpha_d \neq 0. \tag{3.10}$$

Such a choice means in particular that the operation $\oplus_d$ is invertible.

The update filter is taken to be of the form:

$$U_{d_n}(y)(\boldsymbol{n}) = \sum_{j=1}^{N} \lambda_{d_n, j}\, y_j(\boldsymbol{n})\,, \tag{3.11}$$

i.e., it is a linear filter of length N. The filter coefficients $\lambda_{d_n,j}$ depend on the decision $d_{\boldsymbol{n}}$ given by (3.8). Combination of (3.2), (3.10), and (3.11) yields

$$x'(\boldsymbol{n}) = \alpha_{d_n} x(\boldsymbol{n}) + \sum_{j=1}^{N} \beta_{d_n, j}\, y_j(\boldsymbol{n})\,, \tag{3.12}$$

where

$$\beta_{d,j} = \alpha_d \lambda_{d,j} \,.$$

Obviously, we can easily invert (3.12):

$$x(\boldsymbol{n}) = \frac{1}{\alpha_{d_{\boldsymbol{n}}}}\bigl(x'(\boldsymbol{n}) - \sum_{j=1}^{N} \beta_{d_{\boldsymbol{n}},j} y_j(\boldsymbol{n})\bigr) \,,$$

presumed that the decision $d_{\boldsymbol{n}}$ is known. Since $d_{\boldsymbol{n}}$ depends on the components $v_j(\boldsymbol{n}) = x(\boldsymbol{n}) - y_j(\boldsymbol{n})$, $j = 1, \ldots, N$, and $x(\boldsymbol{n})$ is not available at synthesis, recovery of $d_{\boldsymbol{n}}$ from x' and $y = \{y(\cdot|1), \ldots, y(\cdot|P)\}$ is not always possible. Thus, perfect reconstruction is tantamount to the recovery of $d_{\boldsymbol{n}}$, for every location $\boldsymbol{n}$, from x' and y. Define the value

$$\kappa_d = \alpha_d + \sum_{j=1}^{N} \beta_{d,j}, \qquad d = 0, 1 \,.$$

We have the following result.

Proposition 3.4.1. *Assume that the seminorm p on $\mathbb{R}^N$ satisfies the adaptivity condition in* (3.9). *A necessary condition for perfect reconstruction is $\kappa_0 = \kappa_1$.*

Proof. Assume that $\kappa_0 \neq \kappa_1$. Let $\xi \in \mathbb{R}$ be such that

$$|(\kappa_0 - \kappa_1)\xi| > \frac{\alpha_1 T}{p(\boldsymbol{u})} \,. \tag{3.13}$$

Let $\boldsymbol{n}$ be a given location and assume that $x(\boldsymbol{n}) = \xi$ and $y_j(\boldsymbol{n}) = \xi$ for $j = 1, \ldots, N$. Obviously, $\boldsymbol{v}(\boldsymbol{n}) = \boldsymbol{0}$ hence $d_{\boldsymbol{n}} = 0$. It follows immediately that (3.12) gives $x'(\boldsymbol{n}) = \kappa_0 \xi$. However, if we take $x(\boldsymbol{n}) = \xi + \eta$ and the same $y_j(\boldsymbol{n})$ as before, then $\boldsymbol{v}(\boldsymbol{n}) = \eta \boldsymbol{u}$. Therefore, if $|\eta| > T/p(\boldsymbol{u})$, then $d_{\boldsymbol{n}} = 1$ and we deduce that $x'(\boldsymbol{n}) = \kappa_1 \xi + \alpha_1 \eta$. If we choose $\eta = (\kappa_0 - \kappa_1)\xi/\alpha_1$, then, because of (3.13), the condition $|\eta| > T/p(\boldsymbol{u})$ is satisfied. For this particular choice, however, $\kappa_1 \xi + \alpha_1 \eta = \kappa_0 \xi$. Thus, we have shown that for the same values of $y_j(\boldsymbol{n})$, two different inputs for $x(\boldsymbol{n})$ may yield the same output. Clearly, perfect reconstruction is out of reach in such a case. □

Henceforth we assume $\kappa_0 = \kappa_1$. Obviously, to guarantee true adaptivity we need that the update filters for $d = 0$ and $d = 1$ are different.

Adaptivity Condition for the Update Filters. The update filters for $d = 0$ and $d = 1$ do not coincide, i.e.,

$$\beta_{0,j} \neq \beta_{1,j} \quad \text{for at least one } j \in \{1, \ldots, N\} \,. \tag{3.14}$$

Throughout the remainder of this chapter we normalize the filter coefficients so that

$$\kappa_0 = \kappa_1 = 1 \,. \tag{3.15}$$

Note that such a normalization is possible only in the case where $\kappa_d \neq 0$. A system with $\kappa_d = 0$ would, in general, correspond to a prediction operator (i.e., high-pass filtering of x_0 to obtain the detail signal y'), while the condition $\kappa_d \neq 0$ is more appropriate for an update operator (i.e., low-pass filtering of x_0 to obtain the approximation signal x').

Unfortunately, the condition in (3.15) is far from being a sufficient condition for perfect reconstruction. In the following section we will be concerned with the derivation of sufficient conditions for perfect reconstruction.

Henceforth, to simplify notation, we will often omit the argument $\boldsymbol{n}$. Thus we write x, y_j instead of $x(\boldsymbol{n}), y_j(\boldsymbol{n})$, respectively, and $\boldsymbol{v} = (v_1, \ldots, v_N)^T$ instead of $\boldsymbol{v}(\boldsymbol{n}) = \big(v_1(\boldsymbol{n}), \ldots, v_N(\boldsymbol{n})\big)^T$. Now, the update lifting step in (3.12) can be written as

$$x' = \alpha_d x + \sum_{j=1}^{N} \beta_{d,j} y_j \,. \tag{3.16}$$

Subtraction of y_i at both sides of (3.16) yields

$$v_i' = (1 - \beta_{d,i}) v_i - \sum_{j \neq i} \beta_{d,j} v_j \,, \tag{3.17}$$

where

$$v_i' = x' - y_i \,, \qquad i = 1, \ldots, N \,. \tag{3.18}$$

We call $\boldsymbol{v}' = (v_1', \ldots, v_N')^T$ the gradient vector at synthesis, and define the $N \times N$ matrix A_d by the right hand-side expression in (3.17), i.e.,

$$A_d = \begin{pmatrix} 1-\beta_{d,1} & -\beta_{d,2} & -\beta_{d,3} & \cdots & -\beta_{d,N} \\ -\beta_{d,1} & 1-\beta_{d,2} & -\beta_{d,3} & \cdots & \vdots \\ -\beta_{d,1} & -\beta_{d,2} & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ -\beta_{d,1} & -\beta_{d,2} & -\beta_{d,3} & \cdots & 1-\beta_{d,N} \end{pmatrix} . \tag{3.19}$$

The adaptive update lifting step is described therefore by

$$\begin{cases} \boldsymbol{v}' = A_d \boldsymbol{v} \\ d = [p(\boldsymbol{v}) > T] \,, \end{cases}$$

where p: $\mathbb{R}^N \to \mathbb{R}_+$ is a given seminorm satisfying the adaptivity condition (3.9). In addition, we assume that the adaptivity condition (3.14) for the filters is satisfied, hence $A_0 \neq A_1$.

Note that the matrix A_d can also be written as

$$A_d = I - \boldsymbol{u} \boldsymbol{\beta}_d^T \,, \tag{3.20}$$

where I is the $N \times N$ identity matrix, and $\boldsymbol{u} = (1, \ldots, 1)^T$ and $\boldsymbol{\beta}_d = (\beta_{d,1}, \ldots, \beta_{d,N})^T$ are column vectors of length N. For its determinant we find, after simple algebraic manipulations,

$$\det(A_d) = 1 - \boldsymbol{\beta}_d^T \boldsymbol{u} = 1 - \sum_{j=1}^{N} \beta_{d,j} = \alpha_d \,,$$

where we have used (3.15). Since we have assumed that $\alpha_d \neq 0$ for $d = 0, 1$, we may conclude that A_d is invertible. Moreover, one can easily show that

$$A_d^{-1} = I + \frac{1}{\alpha_d} \boldsymbol{u} \boldsymbol{\beta}_d^T = \begin{pmatrix} 1 + \frac{\beta_{d,1}}{\alpha_d} & \frac{\beta_{d,2}}{\alpha_d} & \frac{\beta_{d,3}}{\alpha_d} & \cdots & \frac{\beta_{d,N}}{\alpha_d} \\ \frac{\beta_{d,1}}{\alpha_d} & 1 + \frac{\beta_{d,2}}{\alpha_d} & \frac{\beta_{d,3}}{\alpha_d} & \cdots & \vdots \\ \vdots & \frac{\beta_{d,2}}{\alpha_d} & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ \frac{\beta_{d,1}}{\alpha_d} & \cdots & \cdots & \cdots & 1 + \frac{\beta_{d,N}}{\alpha_d} \end{pmatrix}.$$

Putting

$$\boldsymbol{\beta}'_d = -\boldsymbol{\beta}_d / \alpha_d \,,$$

we find that A_d^{-1} takes a form similar to that of A_d:

$$A_d^{-1} = I - \boldsymbol{u} \boldsymbol{\beta}_d'^T \,.$$

These expressions will be useful in the derivation of perfect reconstruction conditions.

3.5 When do we have perfect reconstruction?

In this section we formulate conditions on the seminorm and the update filters which guarantee perfect reconstruction. As a preparatory step, we will 'translate' the perfect reconstruction condition into another condition called the *threshold criterion*, stated in terms of the seminorm.

Recall that the update lifting step described in the previous section is given by

$$\begin{cases} \boldsymbol{v}' = A_d \boldsymbol{v} \\ d = [p(\boldsymbol{v}) > T] \,. \end{cases}$$

If $p(\boldsymbol{v}) \leq T$ at the analysis step, then the decision equals $d = 0$ and $\boldsymbol{v}' = A_0 \boldsymbol{v}$. If, on the other hand, $p(\boldsymbol{v}) > T$, then $d = 1$ and $\boldsymbol{v}' = A_1 \boldsymbol{v}$. To have perfect reconstruction we must be able to recover the decision d from the gradient vector at synthesis $\boldsymbol{v}'$. For simplicity, we shall restrict ourselves to the case where d can be recovered by thresholding the seminorm $p(\boldsymbol{v}')$, i.e., the case that

$$d = [p(\boldsymbol{v}) > T] = [p(\boldsymbol{v}') > T'] \,,$$

for some $T' > 0$. We formalize this condition in the following criterion.

Threshold Criterion. Given a threshold $T > 0$, there exists a (possibly different) threshold $T' > 0$ such that

(*i*) if $p(\boldsymbol{v}) \leq T$ then $p(A_0 \boldsymbol{v}) \leq T'$;

(*ii*) if $p(\boldsymbol{v}) > T$ then $p(A_1 \boldsymbol{v}) > T'$.

The following result is obvious.

Proposition 3.5.1. *If the threshold criterion holds then we have perfect reconstruction.*

The corresponding reconstruction algorithm is straightforward:

1. compute $\boldsymbol{v}'$ from (3.18);
2. if $p(\boldsymbol{v}') \le T'$ then $d = 0$, otherwise $d = 1$;
3. compute x from (3.16), i.e.,

$$x = \frac{1}{\alpha_d}(x' - \sum_{j=1}^{N} \beta_{d,j} y_j)\,.$$

Thus it remains to verify the validity of the threshold criterion. The following result provides necessary and sufficient conditions.

Proposition 3.5.2. *The threshold criterion holds if and only if the following three conditions are satisfied:*

$$p(A_0) < \infty \text{ and } p^{-1}(A_1) < \infty \tag{3.21}$$

$$p(A_0)p^{-1}(A_1) \le 1\,. \tag{3.22}$$

Proof. In this proof, we use the first and the second threshold criterion conditions defined in page 55, denoted by (i) and (ii) respectively.

'if': put $T' = p(A_0)T$; we show that the threshold criterion holds. To prove (i), assume that $p(\boldsymbol{v}) \le T$. If $p(\boldsymbol{v}) = 0$, then $p(A_0\boldsymbol{v}) = 0$ by (3.21) and Proposition 3.3.2(b). If $p(\boldsymbol{v}) > 0$, then we get from Proposition 3.3.2(d) that

$$p(A_0\boldsymbol{v}) \le p(A_0)p(\boldsymbol{v}) \le p(A_0)T = T'\,.$$

To prove (ii) assume that $p(\boldsymbol{v}) > T$. From the fact that $p^{-1}(A_1) < \infty$ and Proposition 3.3.2(c) we conclude that $p(A_1\boldsymbol{v}) \ne 0$ and we get from Proposition 3.3.2(e) that $p(\boldsymbol{v}) \le p^{-1}(A_1)p(A_1\boldsymbol{v})$. In combination with (3.22), this gives us

$$p(A_1\boldsymbol{v}) \ge \frac{p(\boldsymbol{v})}{p^{-1}(A_1)} \ge p(A_0)p(\boldsymbol{v}) > p(A_0)T = T'\,.$$

This concludes the proof of the 'if'-part.

'only if': to prove that $p(A_0) < \infty$, assume that $p(\boldsymbol{v}) = 0$ and $p(A_0\boldsymbol{v}) \ne 0$. We show that this will give rise to a contradiction. Choosing $\lambda > T'/p(A_0\boldsymbol{v})$ we have $|\lambda|p(A_0\boldsymbol{v}) > T'$. However $p(\lambda\boldsymbol{v}) = |\lambda|p(\boldsymbol{v}) = 0$, and we have a contradiction with (i). The fact that $p^{-1}(A_1) < \infty$ is proved analogously. Thus it remains to prove (3.22). Choose $T = 1$ and let T' be the corresponding threshold given by the threshold criterion. We derive from (i) that $p(A_0) \le T'$. Now (ii) reads as follows: if $p(\boldsymbol{v}) > 1$ then $p(A_1\boldsymbol{v}) > T'$. Suppose that (3.22) does not hold, i.e., $p(A_0)p^{-1}(A_1) > 1$ ($p(A_0) \ne 0$, otherwise $p^{-1}(A_1)$ should be infinite). From the definition of $p^{-1}(A_1)$ (see page 50), it follows that there must be a vector $\boldsymbol{v} \in \mathbb{R}^N$ with $p(A_1\boldsymbol{v}) = 1$ and $p(A_0)p(\boldsymbol{v}) > 1$. Putting $\boldsymbol{v}' = p(A_0)\boldsymbol{v}$, we get $p(\boldsymbol{v}') > 1$ and $p(A_1\boldsymbol{v}') = p(A_0) \le T'$ which contradicts (ii). Therefore, (3.22) must hold. □

Note that the proof of the above proposition shows that it is sufficient to choose $T' = p(A_0)T$.

We have shown that a sufficient condition for perfect reconstruction is the threshold criterion, i.e., (3.21)-(3.22). In the next chapter, we will specialize to certain class of seminorms. Now, we prove some results related to the specific form of the matrix $A_d = I - \boldsymbol{u}\boldsymbol{\beta}_d^T$. We start with the following auxiliary result.

Proposition 3.5.3. *Let p be a seminorm on $\mathbb{R}^N$ and let V_0 be the kernel of p, i.e., the linear subspace of $\mathbb{R}^N$ given by*

$$V_0 = \{\boldsymbol{v} \in \mathbb{R}^N \mid p(\boldsymbol{v}) = 0\}.$$

If $A = I - \boldsymbol{u}\boldsymbol{\beta}^T$ and $p(\boldsymbol{u}) \neq 0$, then $p(A) < \infty$ if and only if $\boldsymbol{\beta} \in V_0^\perp$.

Proof. 'if': assume that $\boldsymbol{\beta} \in V_0^\perp$. Following Proposition 3.3.2 we must show that $p(\boldsymbol{v}) = 0$ implies that $p(A\boldsymbol{v}) = 0$. If $p(\boldsymbol{v}) = 0$ then $\boldsymbol{v} \in V_0$ hence $\boldsymbol{\beta}^T\boldsymbol{v} = 0$. This implies that $A\boldsymbol{v} = \boldsymbol{v} - \boldsymbol{u}\boldsymbol{\beta}^T\boldsymbol{v} = \boldsymbol{v}$ and hence that $p(A\boldsymbol{v}) = 0$.

'only if': assume that $p(A) < \infty$ and $\boldsymbol{\beta} \notin V_0^\perp$. Thus there is a $\boldsymbol{v} \in V_0$ with $\boldsymbol{\beta}^T\boldsymbol{v} = 1$. Then $A\boldsymbol{v} = \boldsymbol{v} - \boldsymbol{u}\boldsymbol{\beta}^T\boldsymbol{v} = \boldsymbol{v} - \boldsymbol{u}$. Since $p(\boldsymbol{u}) \neq 0$, we have $0 \neq p(\boldsymbol{u}) \leq p(\boldsymbol{u} - \boldsymbol{v}) + p(\boldsymbol{v}) = p(\boldsymbol{u} - \boldsymbol{v})$, and therefore $p(\boldsymbol{u} - \boldsymbol{v}) = p(\boldsymbol{v} - \boldsymbol{u}) = p(A\boldsymbol{v}) \neq 0$. Since $p(\boldsymbol{v}) = 0$ we conclude from Proposition 3.3.2 that $p(A) = \infty$, a contradiction. This concludes the proof. □

We now investigate the eigenvalue problem $A\boldsymbol{v} = \lambda\boldsymbol{v}$ with $A = I - \boldsymbol{u}\boldsymbol{\beta}^T$. This can be written as $\boldsymbol{v} - \boldsymbol{u}\boldsymbol{\beta}^T\boldsymbol{v} = \lambda\boldsymbol{v}$. We have to distinguish the cases $\lambda = 1$ and $\lambda \neq 1$. If $\lambda = 1$ we find $\boldsymbol{\beta}^T\boldsymbol{v} = 0$ and from $\lambda \neq 1$ we get that $\boldsymbol{v}$ is a multiple of $\boldsymbol{u}$. Thus we arrive at the following result.

Lemma 3.5.4. *Let $A = I - \boldsymbol{u}\boldsymbol{\beta}^T$ and $\alpha = \det(A) = 1 - \boldsymbol{\beta}^T\boldsymbol{u}$.*

(*a*) *If $\alpha = 1$ then A has only one eigenvalue $\lambda = 1$; the eigenspace is the hyperplane $\boldsymbol{\beta}^T\boldsymbol{v} = 0$.*

(*b*) *If $\alpha \neq 1$ then A has eigenvalues $1, \alpha$. The eigenspace associated with eigenvalue $\lambda = 1$ is the hyperplane $\boldsymbol{\beta}^T\boldsymbol{v} = 0$, and the eigenvector associated with $\lambda = \alpha$ is $\boldsymbol{u}$.*

Note that in both the cases (*a*) and (*b*) we have $A\boldsymbol{u} = \alpha\boldsymbol{u}$. We apply this result to the matrix A_d given by (3.19) or, alternatively, by (3.20). Assuming $p(\boldsymbol{u}) > 0$ (see (3.9)), we get that

$$\begin{aligned} p(A_d) &\geq p(A_d\boldsymbol{u})/p(\boldsymbol{u}) = |\alpha_d| \\ p^{-1}(A_d) &\geq p(\boldsymbol{u})/p(A_d\boldsymbol{u}) = |\alpha_d|^{-1}. \end{aligned}$$

On the other hand, if there exists a $\boldsymbol{v}$ with $\boldsymbol{\beta}_d^T\boldsymbol{v} = 0$ and $p(\boldsymbol{v}) \neq 0$ then

$$p(A_d) \geq 1 \text{ and } p^{-1}(A_d) \geq 1. \tag{3.23}$$

Thus we arrive at the following necessary conditions for the threshold condition to hold.

Proposition 3.5.5. *Assume that the seminorm p satisfies the adaptivity condition $p(\boldsymbol{u}) > 0$.*
(*a*) *The threshold criterion can only be satisfied if $|\alpha_0| \leq |\alpha_1|$.*
(*b*) *Assume in addition that $p(\boldsymbol{v}_0) \neq 0$, $p(\boldsymbol{v}_1) \neq 0$ for some vectors $\boldsymbol{v}_d$ with $\boldsymbol{\beta}_d^T\boldsymbol{v}_d = 0$ for $d = 0, 1$, then the threshold criterion can only be satisfied if $|\alpha_0| \leq 1 \leq |\alpha_1|$.*

Proof. The threshold criterion can only hold if (3.22) is satisfied, that is $p(A_0)p^{-1}(A_1) \leq 1$. If $p(\boldsymbol{u}) > 0$, then we have $p(A_0) \geq |\alpha_0|$ and $p^{-1}(A_1) \geq |\alpha_1|^{-1}$. Thus a necessary condition for (3.22) to be satisfied is $|\alpha_0| \cdot |\alpha_1|^{-1} \leq 1$. This proves (a).

To prove (b), assume that for $d = 0, 1$ we have $p(\boldsymbol{v}_d) \neq 0$ for some $\boldsymbol{v}_d$ with $\boldsymbol{\beta}_d^T \boldsymbol{v}_d = 0$. Since both $p(A_0)$ and $p^{-1}(A_1)$ are at least 1 by (3.23), we conclude that $|\alpha_0| \leq 1$ and $|\alpha_1|^{-1} \leq 1$. This concludes the proof. □

Before considering a number of special cases in the next chapter, we observe that the problem becomes trivial if $N = 1$. In this case there is, apart from a multiplicative constant, only one seminorm, namely $p(v) = |v|$. Now the threshold criterion holds if and only if $|\alpha_0| \leq |\alpha_1|$. Henceforth we restrict ourselves to the case $N > 1$.

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 hereof 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^∞-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

$$\begin{cases} \boldsymbol{v}' = A_d \boldsymbol{v} \\ d = [p(\boldsymbol{v}) > T] \,, \end{cases}$$

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

$$x' = \alpha_d x + \sum_{j=1}^{N} \beta_{d,j} y_j \,. \tag{4.1}$$

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 $A_d = I - \boldsymbol{u}\boldsymbol{\beta}_d^T$ is invertible, we can rewrite these conditions as

$$p(A_0) < \infty \text{ and } p(A_1^{-1}) < \infty \tag{4.2}$$

$$p(A_0)p(A_1^{-1}) \leq 1 \,. \tag{4.3}$$

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

$$p(\boldsymbol{v}) = \left|\boldsymbol{a}^T \boldsymbol{v}\right| , \quad \text{with } \boldsymbol{a} \neq \boldsymbol{0} \,. \tag{4.4}$$

Since $\boldsymbol{v}$ is a gradient vector, we call this seminorm the *weighted gradient seminorm.* Note that the adaptivity condition $p(\boldsymbol{u}) > 0$ holds if and only if $\boldsymbol{a}^T \boldsymbol{u} \neq 0$, 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 $A = I - \boldsymbol{u}\boldsymbol{\beta}^T$, where $\boldsymbol{u}, \boldsymbol{\beta} \in \mathbb{R}^N$.*

(*a*) $\boldsymbol{a}^T \boldsymbol{u} = 0$ *implies* $p(A) = 1$.

(*b*) *Assume* $\boldsymbol{a}^T\boldsymbol{u} \neq 0$,

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

(*ii*) *if* $\boldsymbol{a}, \boldsymbol{\beta}$ *are not collinear, then* $p(A) = \infty$.

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

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

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

Assume $\boldsymbol{a}^T\boldsymbol{u} = 0$. Then,

$$|\boldsymbol{a}^T A\boldsymbol{v}| = |\boldsymbol{a}^T(I - \boldsymbol{u}\boldsymbol{\beta}^T)\boldsymbol{v}| = |\boldsymbol{a}^T\boldsymbol{v} - \boldsymbol{a}^T\boldsymbol{u}\boldsymbol{\beta}^T\boldsymbol{v}| = 1\,.$$

This proves (*a*).

Now, assume $\boldsymbol{a}^T\boldsymbol{u} \neq 0$. We distinguish two cases, namely $\boldsymbol{\beta}$ and $\boldsymbol{a}$ *are* or *are not* collinear.

(*i*) $\boldsymbol{\beta}$ collinear with $\boldsymbol{a}$. In this case we can write $\boldsymbol{\beta} = \gamma\,\boldsymbol{a}$ for some constant $\gamma \in \mathbb{R}$ and we get

$$|\boldsymbol{a}^T A\boldsymbol{v}| = |\boldsymbol{a}^T(I - \boldsymbol{u}\boldsymbol{\beta}^T)\boldsymbol{v}| = |\boldsymbol{a}^T\boldsymbol{v} - \gamma\boldsymbol{a}^T\boldsymbol{u}\boldsymbol{a}^T\boldsymbol{v}| = |1 - \gamma\boldsymbol{a}^T\boldsymbol{u}||\boldsymbol{a}^T\boldsymbol{v}| = |1 - \boldsymbol{\beta}^T\boldsymbol{u}| = |\alpha|\,.$$

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

(*ii*) $\boldsymbol{\beta}$ not collinear with $\boldsymbol{a}$. In this case we can express $\boldsymbol{\beta} = \gamma\boldsymbol{a} + \boldsymbol{c}$ with $\boldsymbol{a}^T\boldsymbol{c} = 0$ and $\boldsymbol{c} \neq 0$. Let us choose $\boldsymbol{v}$ such that $\boldsymbol{a}^T\boldsymbol{v} = 0$ and $\boldsymbol{c}^T\boldsymbol{v} \neq 0$. Then, $p(\boldsymbol{v}) = |\boldsymbol{a}^T\boldsymbol{v}| = 0$ and

$$p(A\boldsymbol{v}) = |\boldsymbol{a}^T A\boldsymbol{v}| = |\boldsymbol{a}^T\boldsymbol{u}\boldsymbol{c}^T\boldsymbol{v}| \neq 0\,.$$

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

Thus we arrive at the following result.

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

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

(*b*) $\boldsymbol{\beta}_0$, $\boldsymbol{\beta}_1$, $\boldsymbol{a}$ *are collinear and* $|\alpha_0| \leq |\alpha_1|$.

Proof. If $\boldsymbol{a}^T\boldsymbol{u} = 0$, we conclude from Lemma 4.1.1 that $p(A_0) = p(A_1^{-1}) = 1$. This holds independently whether $\boldsymbol{\beta}_d$ and $\boldsymbol{a}$ are collinear. Clearly, conditions in (4.2)-(4.3) are satisfied.

Consider now the case where $\boldsymbol{a}^T\boldsymbol{u} \neq 0$. If $\boldsymbol{\beta}_d$ and $\boldsymbol{a}$ are collinear, the previous lemma yields that $p(A_0) = |\alpha_0|$ and $p(A_1^{-1}) = |\alpha_1|^{-1}$. Thus, from (4.2)-(4.3) we conclude that the threshold criterion holds if and only if $|\alpha_0| \leq |\alpha_1|$.

If $\boldsymbol{\beta}_d$ and $\boldsymbol{a}$ are not collinear, Lemma 4.1.1 yields that $p(A_d) = \infty$ 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 $p(\boldsymbol{v}) = |\boldsymbol{a}^T \boldsymbol{v}|$ 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 $\boldsymbol{\beta}_d = \gamma_d \boldsymbol{a}$ 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 γ_1, γ_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 $\boldsymbol{\beta}_1 = \boldsymbol{0}$; hence $\alpha_1 = 1$. In more homogeneous areas, where $d = 0$, we choose γ_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

$$\gamma_0 = \frac{\sum_j a_j}{\sum_j a_j^2 + \left(\sum_j a_j\right)^2} \tag{4.5}$$

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 $\boldsymbol{v}$ 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 $(1 - \gamma_d \sum_j a_j)^2 + \gamma_d^2 \sum_j a_j^2$.

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 $j = 1, \dots, N$, that is, $\boldsymbol{v}(n) = \big(v_1(n), \dots, v_N(n)\big)^T$. In this subsection, however, we assume that

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

where

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

An illustration is given in Fig. 4.1. With every coefficient vector $\boldsymbol{a} \in \mathbb{R}^{K+L+1}$ in (4.4) we can

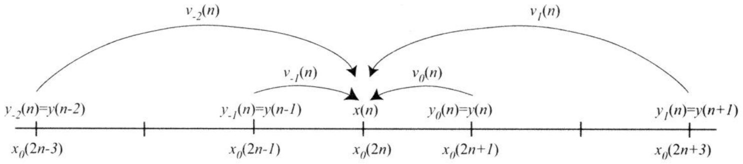


Figure 4.1: *Indexing of the gradient vector.*

associate a filter $\Delta_{\boldsymbol{a}}$ which maps an input vector $\big(y(n-K), \dots, y(n-1), x(n), y(n), \dots, y(n+L)\big)^T$, or equivalently, $\big(x_0(2n-2K+1), \dots, x_0(2n-1), x_0(2n), x_0(2n+1), \dots, x_0(2n+2L+1)\big)^T$ onto an output value

$$\Delta_{\boldsymbol{a}}(x_0)(2n) = \sum_{j=-K}^{L} a_j v_j(n) = \sum_{j=-K}^{L} a_j\big(x_0(2n) - x_0(2n+2j+1)\big)\,. \tag{4.6}$$

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

$$N \le L + K + 1\,.$$

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 $N \ge 1$, the coefficient vectors $\boldsymbol{a} \in \mathbb{R}^{K+L+1}$ for which the corresponding filter $\Delta_{\boldsymbol{a}}$ 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 $n \in \mathbb{Z}$,

$$\sum_{j=-K}^{L} a_j \left[(2n)^k - (2n+2j+1)^k\right] = 0 \quad \text{for} \quad k = 0, \ldots, N-1\,,$$

which is satisfied if and only if either $N = 1$, or $N > 1$ and

$$\sum_{j=-K}^{L} a_j (2j+1)^k = 0 \quad \text{for} \quad k = 1, \ldots, N-1\,.$$

Consider the function $Q_{\boldsymbol{a}}$ given by

$$Q_{\boldsymbol{a}}(z) = \sum_{j=-K}^{L} a_j (1 - z^{2j+1})\,.$$

The proof of the following result is straightforward.

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

We next consider the case $N > 1$. Obviously, $Q_{\boldsymbol{a}}$ has a zero of multiplicity N if and only if $Q'_{\boldsymbol{a}}$ (the derivative of $Q_{\boldsymbol{a}}$ with respect to z) has a zero of multiplicity $N-1$. Now

$$Q'_{\boldsymbol{a}}(z) = -\sum_{j=-K}^{L} a_j (2j+1) z^{2j}\,, \tag{4.7}$$

and if $Q'_{\boldsymbol{a}}$ has a zero at $z = 1$ with multiplicity $N-1$, then we can write

$$Q'_{\boldsymbol{a}}(z) = (z-1)^{N-1} z^{-2K} R(z)\,, \tag{4.8}$$

with $R(1) \neq 0$ and

$$R(z) = \sum_{i=0}^{2(L+K)-N+1} r_i z^i\,.$$

From the fact that $Q'_{\boldsymbol{a}}$ 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

$$R(z) = (z+1)^{N-1} \sum_{i=0}^{L+K+1-N} q_i z^{2i}\,. \tag{4.9}$$

Substitution of (4.9) into (4.8) yields

$$\begin{aligned} Q'_{\boldsymbol{a}}(z) &= z^{-2K}(z^2-1)^{N-1}\sum_{i=0}^{L+K+1-N} q_i z^{2i} \\ &= z^{-2K}\sum_{l=0}^{N-1}\binom{N-1}{l}(-1)^{N-1-l}\sum_{i=0}^{L+K+1-N} q_i z^{2(i+l)} . \end{aligned}$$

Recall that $L+K+1 \geq N > 1$. Replacing the summation variable i by $j = i + l$, we get

$$\begin{aligned} Q'_{\boldsymbol{a}}(z) &= z^{-2K}\sum_{j=0}^{L+K}\sum_{l=\max\{0,j-L-K-1+N\}}^{\min\{N-1,j\}}\binom{N-1}{l}(-1)^{N-1-l}q_{j-l}z^{2j} \\ &= \sum_{j=-K}^{L}\sum_{l=\max\{0,j-L-1+N\}}^{\min\{N-1,j+K\}}\binom{N-1}{l}(-1)^{N-1-l}q_{j+K-l}z^{2j} . \end{aligned}$$

In combination with (4.7), this yields the following expression for the coefficients a_j:

$$-(2j+1)a_j = \sum_{l=\max\{0,j-L-1+N\}}^{\min\{N-1,j+K\}}\binom{N-1}{l}(-1)^{N-1-l}q_{j+K-l} .$$

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 $q_0 = -1$)

$$a_j = \frac{(-1)^{L+j}}{2j+1}\binom{2L}{L+j}, \tag{4.10}$$

and if $K = L+1$ and $N = 2L+2$ (even-length filter), we get

$$a_j = \frac{(-1)^{L+j}}{2j+1}\binom{2L+1}{L+j+1} . \tag{4.11}$$

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 $\boldsymbol{a} \in \mathcal{A}_N$, then the corresponding decision map does not respond to polynomials up to degree $N-1$, i.e., the corresponding expression $|\boldsymbol{a}^T\boldsymbol{v}(n)|$ 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 $\boldsymbol{a} = (a_{-1}, a_0)^T = (1,1)^T$. From Proposition 4.1.2 we conclude that the threshold criterion holds if $\boldsymbol{\beta}_0 = \gamma_0(1,1)^T$ and $\boldsymbol{\beta}_1 = \gamma_1(1,1)^T$ with

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

Choosing γ_1 and γ_0 as in Example 4.1.3, we get $\alpha_1 = 1$, $\boldsymbol{\beta}_1 = \mathbf{0}$ and $\alpha_0 = \frac{1}{3}$, $\boldsymbol{\beta}_0 = \frac{1}{3}(1,1)^T$.

4.1.2 Simulations

In this subsection we show some simulation results using the weighted gradient seminorm $p(\boldsymbol{v}) = |\boldsymbol{a}^T\boldsymbol{v}|$. In all cases, we choose filter coefficients α_d, $\boldsymbol{\beta}_d$ such that the threshold criterion holds. Note that given the weight vector $\boldsymbol{a}$, once the parameters γ_0, γ_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 $p(\boldsymbol{u})$.

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 × 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 $\boldsymbol{v}(n)$ 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:

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

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

Experiment 4.1.1 (Seminorm $p(\boldsymbol{v}) = |\boldsymbol{u}^T\boldsymbol{v}|$ for 1D, N = 4 - Fig. 4.2)
First we consider the case where $\boldsymbol{a} = \boldsymbol{u} = (1,1,1,1)^T$. Proposition 4.1.2 yields that we must choose $\boldsymbol{\beta}_d = \gamma_d(1,1,1,1)^T$ for some constants γ_0, γ_1 such that

$$|1 - 4\gamma_0| \le |1 - 4\gamma_1| .$$

For $d = 1$, we choose $\gamma_1 = 0$ and thus $\boldsymbol{\beta}_1 = \mathbf{0}$. For $d = 0$ we choose γ_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[3] 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 $\boldsymbol{a}$ such that the decision map does not respond to polynomial regions of order 3. This gives $\boldsymbol{a} = (-1/3, 3, 3, -1/3)^T$. Choosing $\gamma_1 = 0$ and γ_0 as in (4.5), we get

$$\gamma_0 = 4/35 \text{ and hence } \boldsymbol{\beta_0} = (4/35)\boldsymbol{a}\,.$$

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

---

[3] Recall 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.

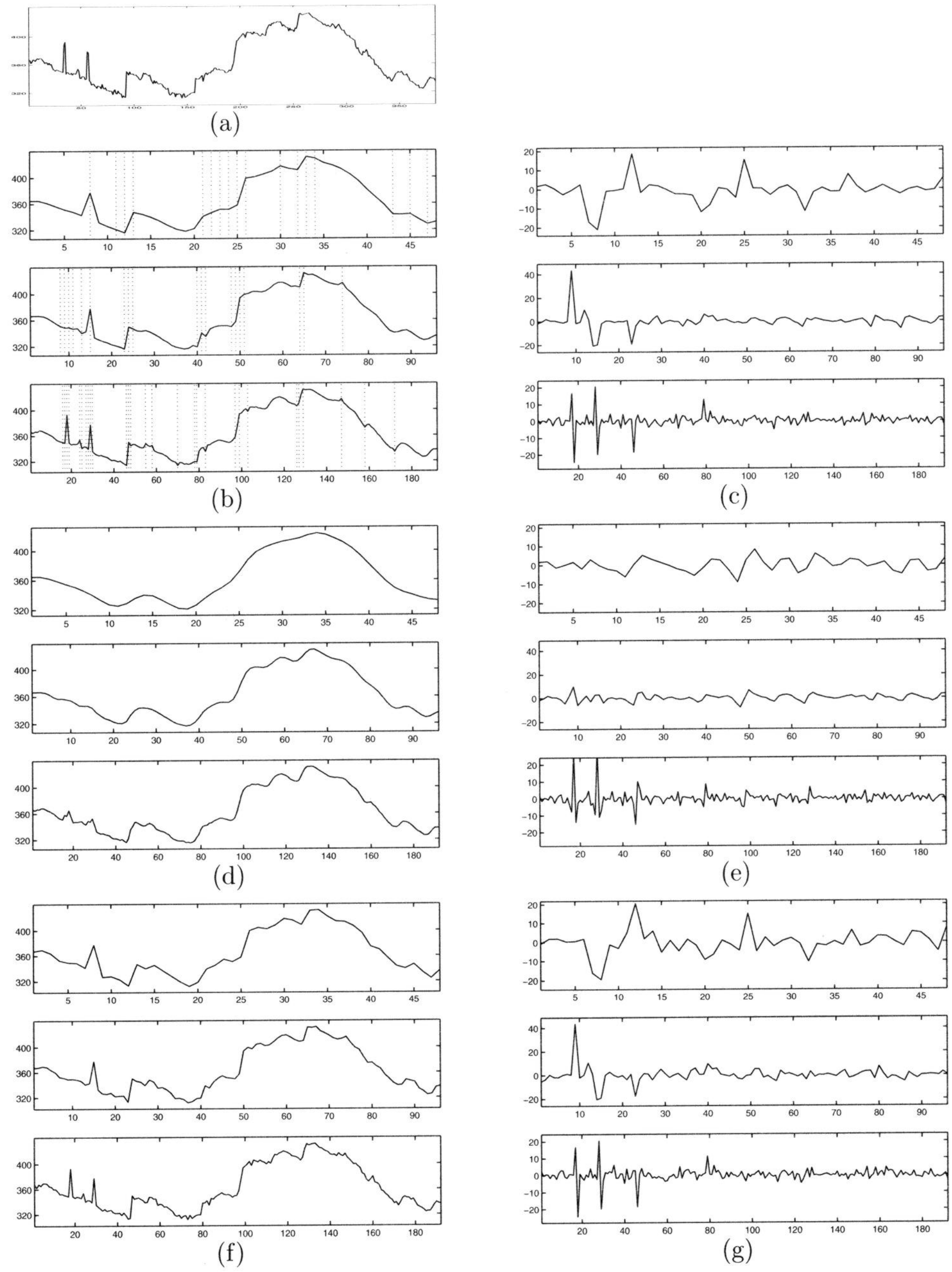
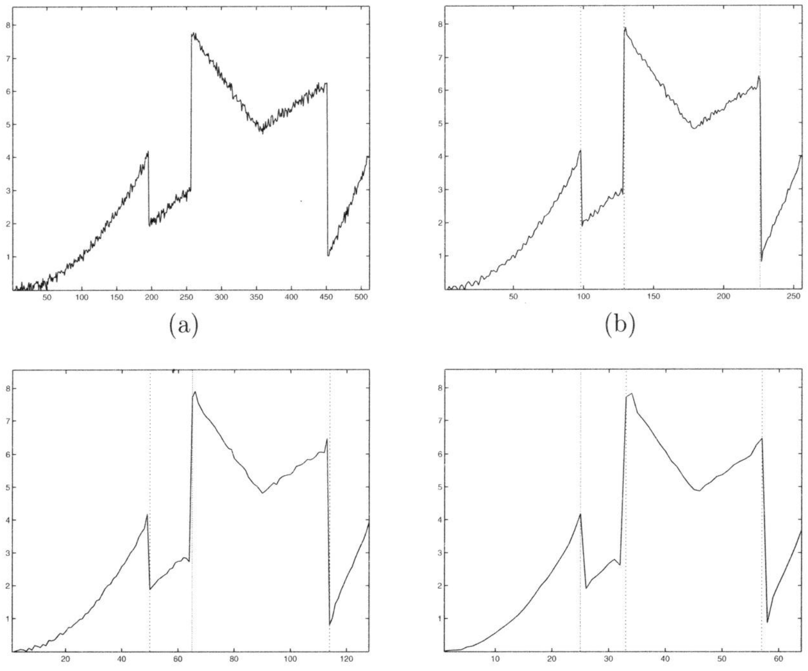

(a) (b) (c) (d) (e) (f) (g)

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* $d = 1$.



(a) (b)

(c) (d)

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

$$y'(m,n) = y(m,n) - \frac{1}{4}\sum_{j=1}^{4} x'_j(m,n)\,, \tag{4.13}$$

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,

$$p(\boldsymbol{v}) = |\sum_{j=1}^{4} v_j|\,.$$

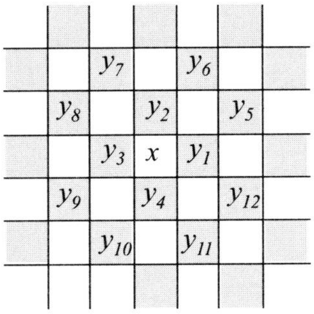


Figure 4.4: *Indexing of samples for a quincunx sampling structure centered at sample x.*

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 $y(\cdot|1), y(\cdot|2), y(\cdot|3)$. 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'_h, y'_v, y'_d with a prediction scheme

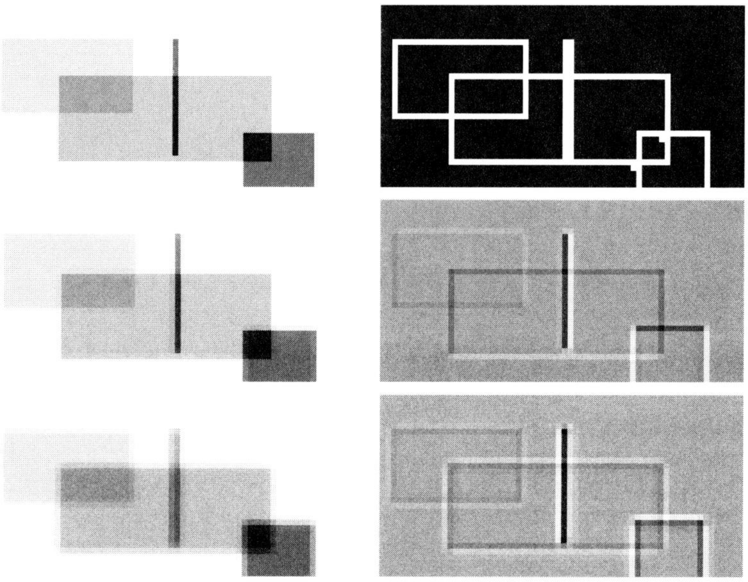


Figure 4.5: *Decompositions (at level 2) corresponding with Experiment 4.1.3. Top: input image (left) and decision map (right) using a threshold* $T = 20$. *Middle: approximation (left) and detail (right) images in the adaptive case. Bottom: approximation (left) and detail (right) images in the non-adaptive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

$x_0(2m-1,2n-1)$ $y_d(m-1,n-1)$ $y_6(m,n)$	$x_0(2m-1,2n)$ $y_h(m-1,n)$ $y_2(m,n)$	$x_0(2m-1,2n+1)$ $y_d(m-1,n)$ $y_5(m,n)$
$x_0(2m,2n-1)$ $y_v(m,n-1)$ $y_3(m,n)$	$x_0(2m,2n)$ $\mathbf{x(m,n)}$	$x_0(2m,2n+1)$ $y_v(m,n)$ $y_1(m,n)$
$x_0(2m+1,2n-1)$ $y_d(m,n-1)$ $y_7(m,n)$	$x_0(2m+1,2n)$ $y_h(m,n)$ $y_4(m,n)$	$x_0(2m+1,2n+1)$ $y_d(m,n)$ $y_8(m,n)$

Figure 4.6: *Indexing of samples in a* 3×3 *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

$$\begin{aligned} y'_h &= y_h - x' && (4.14)\\ y'_v &= y_v - x' && (4.15)\\ y'_d &= y_d - x' - y'_v - y'_h\,. && (4.16) \end{aligned}$$

Alternatively, $y'_d = y_d + x' - y_v - y_h$. Note that the resulting 2D wavelet decomposition is non-separable.

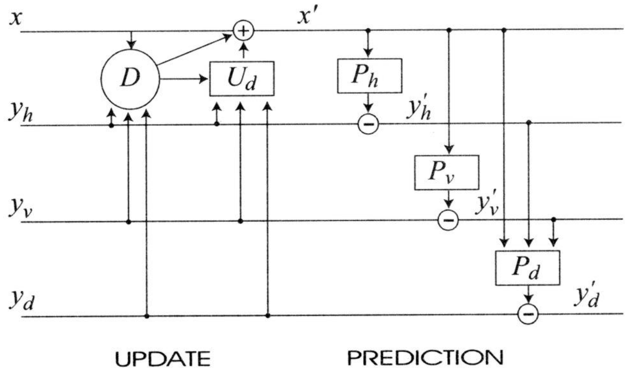


Figure 4.7: *2D wavelet decomposition comprising an adaptive update lifting step (left) and three consecutive (fixed) prediction lifting steps (right).*

Experiment 4.1.4 (Seminorm $p(\boldsymbol{v}) = |\boldsymbol{a}^T\boldsymbol{v}|$ for 2D square, N = 8 - Fig. 4.8)
Consider the seminorm given by

$$p(\boldsymbol{v}) = \left|\boldsymbol{a}^T\boldsymbol{v}\right| = \left|\sum_{j=1}^{8} a_j v_j\right| ,$$

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 $p(\boldsymbol{v})$ does not depend on x. According to Proposition 4.1.2, we choose the filter coefficients

$$\boldsymbol{\beta}_d = \gamma_d \boldsymbol{a} \ \text{ with } \ |1 - \gamma_0 \sum_{j=1}^{8} a_j| \leq |1 - \gamma_1 \sum_{j=1}^{8} a_j|\,.$$

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 $p(\boldsymbol{v})$ 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 $a_j = 0$ 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 γ_0, γ_1 such that $|1 - 2\gamma_0| \leq |1 - 2\gamma_1|$. In this experiment we consider this latter choice of $\boldsymbol{a}$, i.e., $\boldsymbol{a} = (1, 1, 1, 1, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2})^T$, 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 $\Delta_{\boldsymbol{a}}$ 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:

$$p(\boldsymbol{v}) = \left(\boldsymbol{v}^T M \boldsymbol{v}\right)^{1/2}, \quad \boldsymbol{v} \in \mathbb{R}^N, \tag{4.17}$$

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 $p(\boldsymbol{u}) = N^{1/2}$, 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* $p_2(\boldsymbol{v}) = \|\boldsymbol{v}\| = (v_1^2 + \cdots + v_N^2)^{\frac{1}{2}}$ *and let* A *be the matrix* $A = I - \boldsymbol{u}\boldsymbol{\beta}^T$, *where* $\boldsymbol{u}, \boldsymbol{\beta} \in \mathbb{R}^N$.

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

(*b*) *If* $\boldsymbol{u}, \boldsymbol{\beta}$ *are not collinear, then* $p_2(A) > 1$.

Proof. (*a*) If $\boldsymbol{u}, \boldsymbol{\beta}$ are collinear, i.e., $\boldsymbol{\beta} = \mu\boldsymbol{u}$ for some constant $\mu \in \mathbb{R}$, then the matrix $A = I - \mu\boldsymbol{u}\boldsymbol{u}^T$ 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 α. Thus $p_2(A) = \max\{1, |\alpha|\}$.

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

$$A\boldsymbol{c} = (I - \boldsymbol{u}\boldsymbol{\beta}^T)\boldsymbol{c} = \boldsymbol{c} - \boldsymbol{u}(\mu\boldsymbol{u} + \boldsymbol{c})^T\boldsymbol{c} = \boldsymbol{c} - (\boldsymbol{c}^T\boldsymbol{c})\boldsymbol{u} = \boldsymbol{c} - \|\boldsymbol{c}\|^2\boldsymbol{u},$$

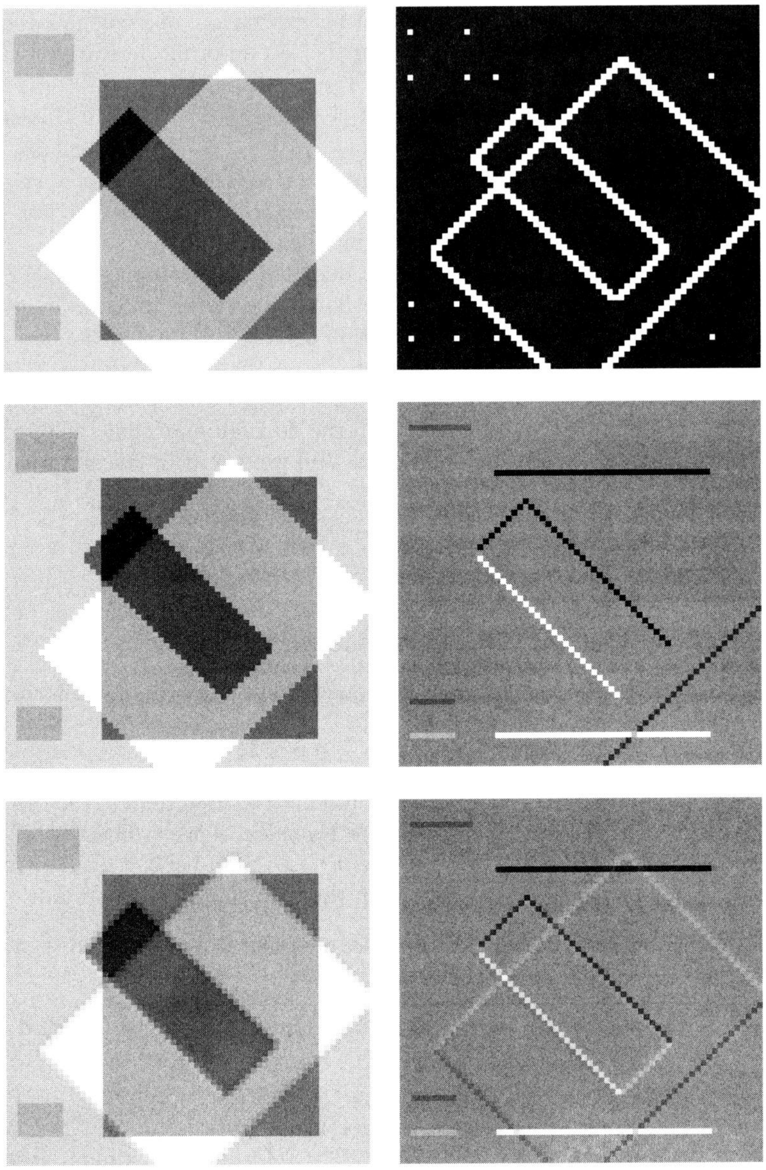


Figure 4.8: *Decompositions (at level 1) corresponding with Experiment 4.1.4. Top: input image (left) and decision map (right) using a threshold $T = 10$. Middle: approximation (left) and horizontal detail (right) images in the adaptive case. Bottom: approximation (left) and horizontal detail (right) images in the non-adaptive case with $d = 0$.*

whence we get that $\|A\boldsymbol{c}\|^2 = \|\boldsymbol{c}\|^2 + N\|\boldsymbol{c}\|^4$, where we have used that $\|\boldsymbol{u}\|^2 = N$. Therefore

$$p_2(A) \geq \|A\boldsymbol{c}\|/\|\boldsymbol{c}\| = (1 + N\|\boldsymbol{c}\|^2)^{\frac{1}{2}} > 1\,,$$

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 $\boldsymbol{u}, \boldsymbol{\beta}_0, \boldsymbol{\beta}_1$ are collinear and $|\alpha_0| \leq 1 \leq |\alpha_1|$.*

Proof. We have $A_0 = I - \boldsymbol{u}\boldsymbol{\beta}_0^T$ and $A_1^{-1} = I - \boldsymbol{u}{\boldsymbol{\beta}'_1}^T$ where $\boldsymbol{\beta}'_1 = -\alpha_1^{-1}\boldsymbol{\beta}_1$. 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 $\boldsymbol{u}, \boldsymbol{\beta}_0, \boldsymbol{\beta}_1$ 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 $1 - {\boldsymbol{\beta}'_1}^T\boldsymbol{u} = 1 + \alpha_1^{-1}(1 - \alpha_1) = \alpha_1^{-1}$. 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

$$M = Q\Lambda Q^T\,, \tag{4.18}$$

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

$$n = \operatorname{rank}(M) = \operatorname{rank}(\Lambda) \leq N\,.$$

Without loss of generality we can assume that

$$\Lambda = \begin{pmatrix} \Lambda_{11} & 0 \\ 0 & 0 \end{pmatrix}, \tag{4.19}$$

where Λ_{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

$$Q = (Q_1\ Q_2)\,, \tag{4.20}$$

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 Λ_{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,*

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

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

In particular, if $\operatorname{rank}(M) = \operatorname{rank}(\Lambda) = N$*, then*

$$p(A) = \|\Lambda^{\frac{1}{2}} Q^T A Q \Lambda^{-\frac{1}{2}}\|. \tag{4.22}$$

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

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

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

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

Decomposing $\boldsymbol{w} = (\boldsymbol{w}_1 \ \boldsymbol{w}_2)^T$, with $\boldsymbol{w}_1 \in \mathbb{R}^n$ and $\boldsymbol{w}_2 \in \mathbb{R}^{N-n}$, we get

$$\boldsymbol{w}^T B^T \Lambda B \boldsymbol{w} = \boldsymbol{w}_1^T B_{11}^T \Lambda_{11} B_{11} \boldsymbol{w}_1 + 2\boldsymbol{w}_1^T B_{11}^T \Lambda_{11} B_{12} \boldsymbol{w}_2 + \boldsymbol{w}_2^T B_{12}^T \Lambda_{11} B_{12} \boldsymbol{w}_2. \tag{4.23}$$

Furthermore, the constraint $\boldsymbol{w}^T \Lambda \boldsymbol{w} = 1$ amounts to $\boldsymbol{w}_1^T \Lambda_{11} \boldsymbol{w}_1 = 1$. This constraint only involves $\boldsymbol{w}_1$ and not $\boldsymbol{w}_2$. This means that maximization of (4.23) yields ∞ 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

$$\begin{aligned} \big(p(A)\big)^2 &= \max\{\boldsymbol{w}_1^T B_{11}^T \Lambda_{11} B_{11} \boldsymbol{w}_1 | \boldsymbol{w}_1^T \Lambda_{11} \boldsymbol{w}_1 = 1\} \\ &= \max\{\boldsymbol{s}^T \Lambda_{11}^{-\frac{1}{2}} B_{11}^T \Lambda_{11}^{\frac{1}{2}} \Lambda_{11}^{\frac{1}{2}} B_{11} \Lambda_{11}^{-\frac{1}{2}} \boldsymbol{s} \mid \boldsymbol{s}^T \boldsymbol{s} = 1\} \\ &= \max\{\|\Lambda_{11}^{\frac{1}{2}} B_{11} \Lambda_{11}^{-\frac{1}{2}} \boldsymbol{s}\|^2 \mid \|\boldsymbol{s}\|^2 = 1\}, \end{aligned}$$

where we have substituted $\boldsymbol{s} = \Lambda_{11}^{\frac{1}{2}} \boldsymbol{w}_1$. This yields

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

which had to be proved.

Finally, if $\operatorname{rank}(M) = N$ then $\Lambda_{11} = \Lambda$, $Q_1 = Q$ and $Q_2 = 0$, and thus (4.21) reduces to (4.22). ☐

We apply this result to the matrix $A_d = I - \boldsymbol{u}\boldsymbol{\beta}_d^T$, $d = 0, 1$. Then

$$Q_1^T A_d Q_2 = Q_1^T Q_2 - Q_1^T \boldsymbol{u}(Q_2^T \boldsymbol{\beta}_d)^T = -Q_1^T \boldsymbol{u}(Q_2^T \boldsymbol{\beta}_d)^T ,$$

since $Q_1^T Q_2 = 0$ by the orthogonality of Q. Therefore, $Q_1^T A_d Q_2 = 0$ if either (i) $Q_1^T \boldsymbol{u} = \boldsymbol{0}$ or (ii) $Q_2^T \boldsymbol{\beta}_d = \boldsymbol{0}$. In case (i) we have $p(A_0) = p(A_1^{-1}) = \|\Lambda_{11}^{\frac{1}{2}} Q_1^T Q_1 \Lambda_{11}^{-\frac{1}{2}}\| = 1$ and the threshold criterion holds. Note, however, that we have $p(\boldsymbol{u}) = 0$ and consequently the adaptivity condition on the seminorm does not hold in this case. We now consider case (ii) where $Q_2^T \boldsymbol{\beta}_d = \boldsymbol{0}$. We compute $p(A_0)$ and $p(A_1^{-1})$ in this case:

$$p(A_0) = \|\Lambda_{11}^{\frac{1}{2}} Q_1^T (I - \boldsymbol{u}\boldsymbol{\beta}_0^T) Q_1 \Lambda_{11}^{-\frac{1}{2}}\| = \|I - \tilde{\boldsymbol{u}}\tilde{\boldsymbol{\beta}}_0^T\| ,$$

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

$$\tilde{\alpha}_0 = 1 - \boldsymbol{u}^T Q_1 Q_1^T \boldsymbol{\beta}_0 .$$

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

$$\tilde{\alpha}_1 = (1 + \frac{1}{\alpha_1} \boldsymbol{u}^T Q_1 Q_1^T \boldsymbol{\beta}_1)^{-1} .$$

Lemma 4.2.4. *If $Q_1^T \boldsymbol{u} \neq 0$, the following two assertions are equivalent:*

(i) *$M\boldsymbol{u}, \boldsymbol{\beta}_d$ are collinear*

(ii) *$\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{\beta}}_d$ are collinear and $Q_2^T \boldsymbol{\beta}_d = \boldsymbol{0}$.*

Proof. Assume (i). We have $M\boldsymbol{u} \neq 0$ (otherwise $Q_1^T M \boldsymbol{u} = \Lambda_{11} Q_1^T \boldsymbol{u} = 0$) and then $\boldsymbol{\beta}_d = \mu_d M \boldsymbol{u} = \mu_d Q_1 \Lambda_{11} Q_1^T \boldsymbol{u}$, where $\mu_d \in \mathbb{R}$. Since $Q_2^T Q_1 = 0$, we find that $Q_2^T \boldsymbol{\beta}_d = \boldsymbol{0}$. Furthermore, $\tilde{\boldsymbol{\beta}}_d = \Lambda_{11}^{-\frac{1}{2}} Q_1^T \boldsymbol{\beta}_d = \mu_d \Lambda_{11}^{\frac{1}{2}} Q_1^T \boldsymbol{u} = \mu_d \tilde{\boldsymbol{u}}$, where we have used that $Q_1^T Q_1 = I$.

Assume (ii): $Q_2^T \boldsymbol{\beta}_d = \boldsymbol{0}$ is equivalent to $\boldsymbol{\beta}_d \in \operatorname{Ran}(Q_2)^\perp = \operatorname{Ran}(Q_1)$, i.e., $\boldsymbol{\beta}_d = Q_1 \boldsymbol{\xi}_d$, where $\boldsymbol{\xi}_d \in \mathbb{R}^n$. Since $\tilde{\boldsymbol{u}}$ and $\tilde{\boldsymbol{\beta}}_d$ are collinear, we have $\tilde{\boldsymbol{\beta}}_d = \mu_d \tilde{\boldsymbol{u}}$, that is $\Lambda_{11}^{-\frac{1}{2}} Q_1^T \boldsymbol{\beta}_d = \mu_d \Lambda_{11}^{\frac{1}{2}} Q_1^T \boldsymbol{u}$, which yields $\boldsymbol{\xi}_d = \mu_d \Lambda_{11} Q_1^T \boldsymbol{u}$, and hence $\boldsymbol{\beta}_d = \mu_d Q_1 \Lambda_{11} Q_1^T \boldsymbol{u} = \mu_d M \boldsymbol{u}$. This concludes the proof. ☐

Therefore, if $\boldsymbol{\beta}_d = \mu_d M \boldsymbol{u}$ with $\mu_d \in \mathbb{R}$, we get

$$\tilde{\alpha}_0 = 1 - \boldsymbol{u}^T Q_1 Q_1^T \boldsymbol{\beta}_0 = 1 - \mu_0 \boldsymbol{u}^T Q_1 \Lambda_{11} Q_1^T \boldsymbol{u} = 1 - \boldsymbol{u}^T \boldsymbol{\beta}_0 = \alpha_0$$

and

$$\tilde{\alpha}_1 = (1 + \frac{1}{\alpha_1} \boldsymbol{u}^T Q_1 Q_1^T \boldsymbol{\beta}_1)^{-1} = (1 + \frac{1}{\alpha_1} \boldsymbol{u}^T \boldsymbol{\beta}_1)^{-1} = \big(1 + \frac{1}{\alpha_1}(1 - \alpha_1)\big)^{-1} = \alpha_1 ;$$

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 $n = \mathrm{rank}(M) \geq 2$. Then the threshold criterion holds if and only if any of the following two conditions holds:*

(*a*) $Q_1^T \boldsymbol{u} = \boldsymbol{0}$ *(in which case the adaptivity condition $p(\boldsymbol{u}) > 0$ is not satisfied);*

(*b*) $\boldsymbol{\beta}_0, \boldsymbol{\beta}_1, M\boldsymbol{u}$ *are collinear and* $|\alpha_0| \leq 1 \leq |\alpha_1|$.

If $n = 1$, then it follows that $M = \boldsymbol{a}\boldsymbol{a}^T$, where $\boldsymbol{a} \in \mathrm{I\!R}^N$, $\boldsymbol{a} \neq \boldsymbol{0}$. In this case, $p(\boldsymbol{v}) = \left(\boldsymbol{v}^T M \boldsymbol{v}\right)^{1/2} = |\boldsymbol{a}^T \boldsymbol{v}|$ for all $\boldsymbol{v} \in \mathrm{I\!R}^N$, 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 Λ a diagonal matrix with strictly positive entries $M_{jj} = \lambda_j$ for $j = 1, \dots, N$. Note that in this case we can write $p(\boldsymbol{v}) = \left(\boldsymbol{v}^T M \boldsymbol{v}\right)^{1/2}$ as

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

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 μ_0, μ_1 such that $\beta_{d,j} = \mu_d \lambda_j$ for $d = 0, 1$ and $j = 1, \dots, N$, 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

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

for minimizing the variance of the noise in the approximation signal. Here $\sum_j$ denotes summation over all indices j. If we take μ_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 $\lambda_1, \dots, \lambda_n$ strictly positive and $\lambda_{n+1} = \cdots = \lambda_N = 0$. The threshold criterion requires that $\boldsymbol{\beta}_d$ is collinear with $M\boldsymbol{u}$. 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*).

4.2.2 Simulations

In this subsection we show some simulation results using the quadratic seminorm $p(\boldsymbol{v}) = \left(\boldsymbol{v}^T M \boldsymbol{v}\right)^{1/2}$. 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 $\boldsymbol{v}(n)$ indexed as in Fig. 4.1. As before, a fixed prediction of the form $y'(n) = y(n) - \frac{1}{2}\big(x'(n) + x'(n+1)\big)$ 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

$$p(\boldsymbol{v}) = \left(\sum_{j=-2}^{1} \lambda_j v_j^2\right)^{1/2} \quad \text{with weights} \quad (\lambda_{-2}, \lambda_{-1}, \lambda_0, \lambda_1) = (\frac{1}{3}, 1, 1, \frac{1}{3}),$$

or equivalently, $M = \operatorname{diag}\big(\frac{1}{3}, 1, 1, \frac{1}{3}\big)$. Following Example 4.2.6(a), we see that the threshold criterion holds if $\boldsymbol{\beta}_d = \mu_d(\frac{1}{3}, 1, 1, \frac{1}{3})^T$, with

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

We take $\mu_1 = 0$ and compute μ_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 $M = \operatorname{diag}\big(\frac{1}{3}, \frac{1}{2}, 1, 1, \frac{1}{2}, \frac{1}{3}\big)$. 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 μ_0, μ_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

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

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

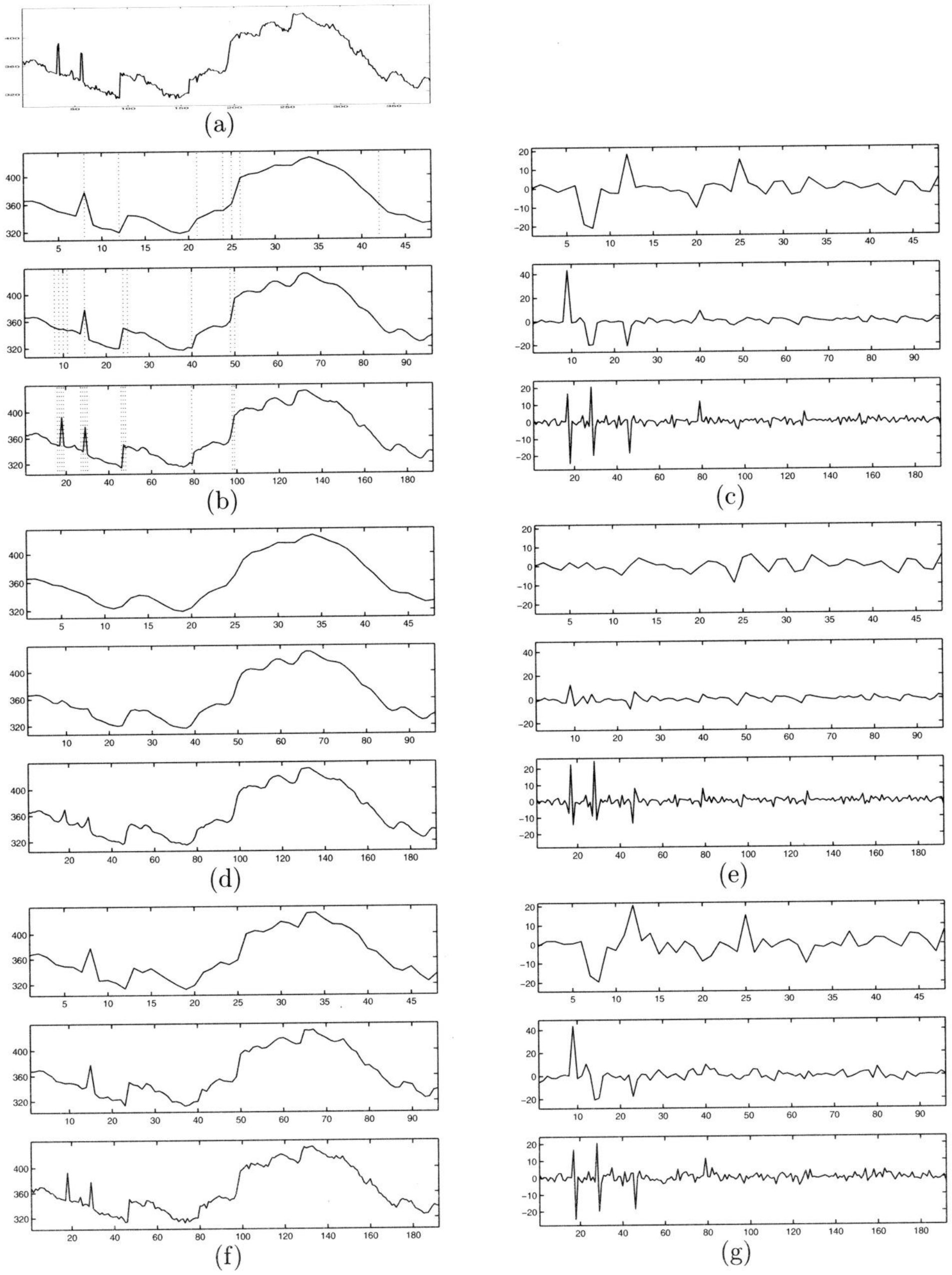


Figure 4.9: *Decompositions (at levels 1, 2 and 3) corresponding with Experiment 4.2.1. (a) Original signal; (b)-(c) approximation and detail signals in the adaptive case using a threshold* $T = 18$*; (d)-(e) approximation and detail signals in the non-adaptive case with* $d = 0$*; (f)-(g) approximation 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.

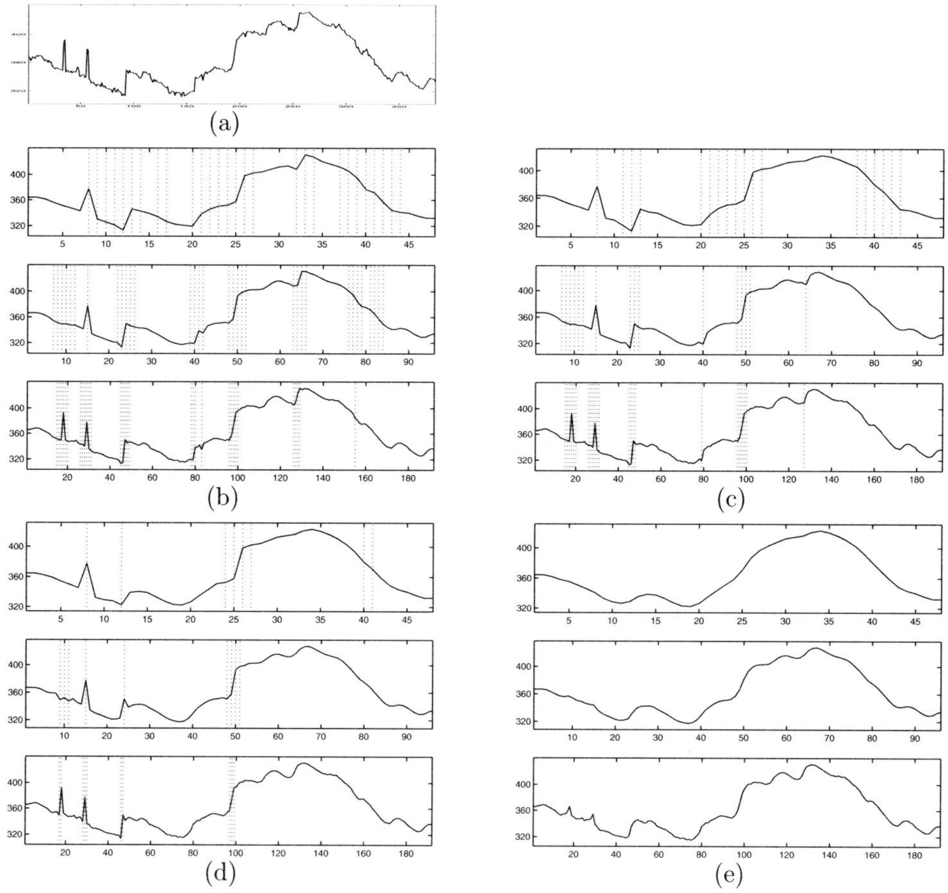


Figure 4.10: *Decompositions (at levels* 1, 2 *and* 3*) corresponding with Experiment 4.2.2. (a) Original signal; (b)-(c) approximation in the adaptive case using a threshold* $T = 15$ *and* $T = 20$*; (d)-(e) approximation in the adaptive case using a threshold* $T = 30$ *and non-adaptive case with*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

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 $j = 1, \dots, 4$, and condition $|\alpha_0| \le 1 \le |\alpha_1|$ reduces to

$$|1 - 4\beta_0| \le 1 \le |1 - 4\beta_1| \,.$$

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 $y_1, \dots, y_{12}$ 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 $M = \mathrm{diag}\big(1,1,1,1,1/\sqrt{5},\dots,1/\sqrt{5}\big)$, and now the formula in (4.25) yields $\mu_0 = \frac{5+2\sqrt{5}}{43+16\sqrt{5}}$. Choosing $\mu_1 = 0$, we have

$$\boldsymbol{\beta}_0 = \mu_0\big(1,1,1,1,\frac{1}{\sqrt{5}},\dots,\frac{1}{\sqrt{5}}\big)^T \quad \text{and} \quad \boldsymbol{\beta}_1 = \mathbf{0}\,.$$

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).

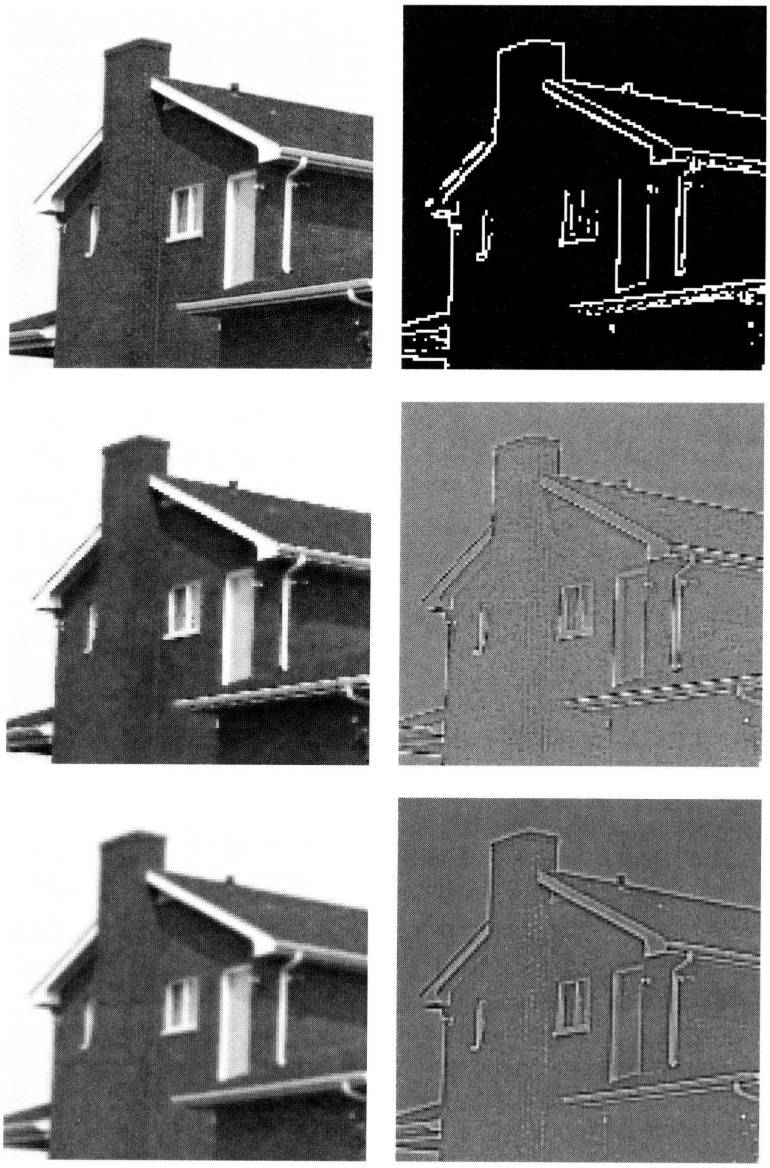


Figure 4.11: *Decompositions (at level 2) corresponding with Experiment 4.2.3. Top: input image (left) and decision map (right) using a threshold* $T = 60$*. Middle: approximation (left) and detail (right) images in the adaptive case. Bottom: approximation (left) and detail (right) images in the non-adaptive case with* $d = 0$*.*

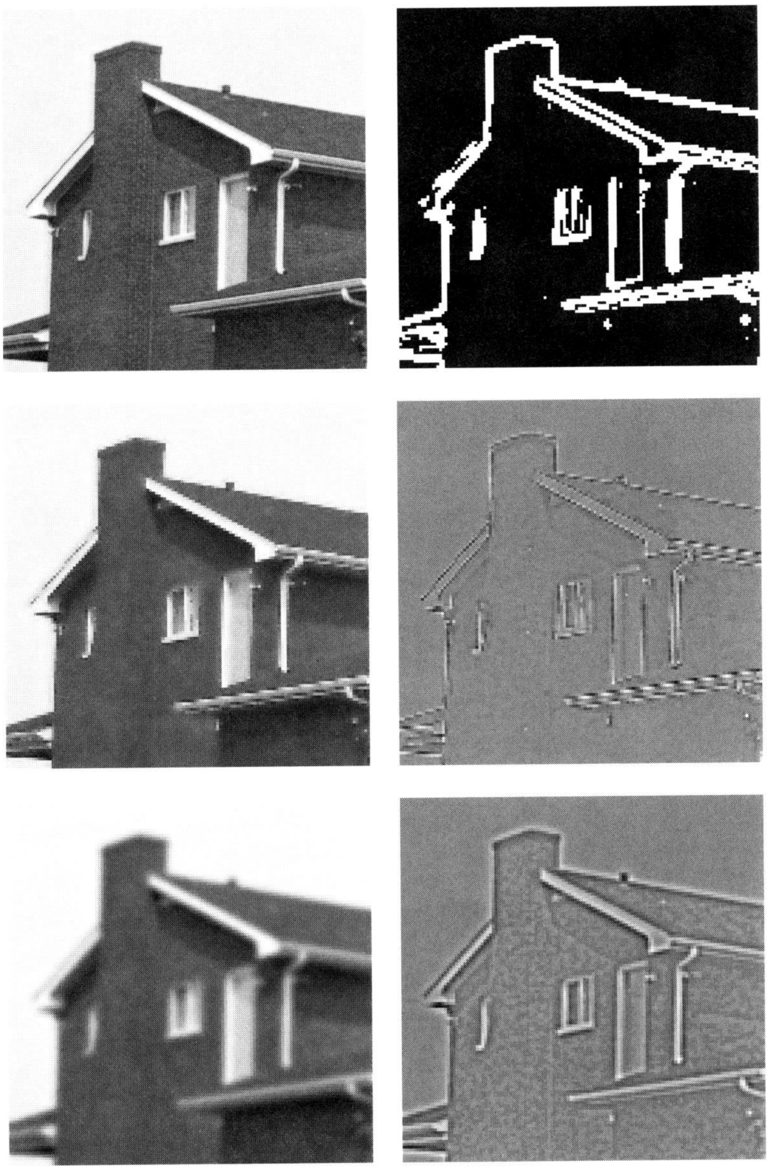


Figure 4.12: *Decompositions (at level* 2*) corresponding with Experiment 4.2.4. Top: input image (left) and decision map (right) using a threshold* $T = 82.5$. *Middle: approximation (left) and detail (right) images in the adaptive case. Bottom: approximation (left) and detail (right) images in the 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

$$p(\boldsymbol{v}) = \left(\sum_{j=1}^{4} |v_j|^2 + \frac{1}{2} (\sum_{j=5}^{8} |v_j|^2) \right)^{1/2} .$$

This corresponds with Example 4.2.6(a) where $M = \mathrm{diag}\big(1,1,1,1,\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2}\big)$. Thus the threshold criterion holds if we choose $\boldsymbol{\beta}_d = \mu_d \left(1,1,1,1,\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2}\right)^T$. As before, we take $\mu_1 = 0$ and compute μ_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^∞-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 $p(\boldsymbol{u}) > 0$ is satisfied. Furthermore, $p(A) < \infty$.

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

$$p_1(\boldsymbol{v}) = \sum_{j=1}^{N} |v_j| ,$$

or the l^∞-norm

$$p_\infty(\boldsymbol{v}) = \max_{j=1,\ldots,N} |v_j| .$$

Recall that $A_d = I - \boldsymbol{u}\boldsymbol{\beta}_d^T$ 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

$$p_1(A_d) = \max_j (|1 - \beta_{d,j}| + (N-1)|\beta_{d,j}|) ,$$

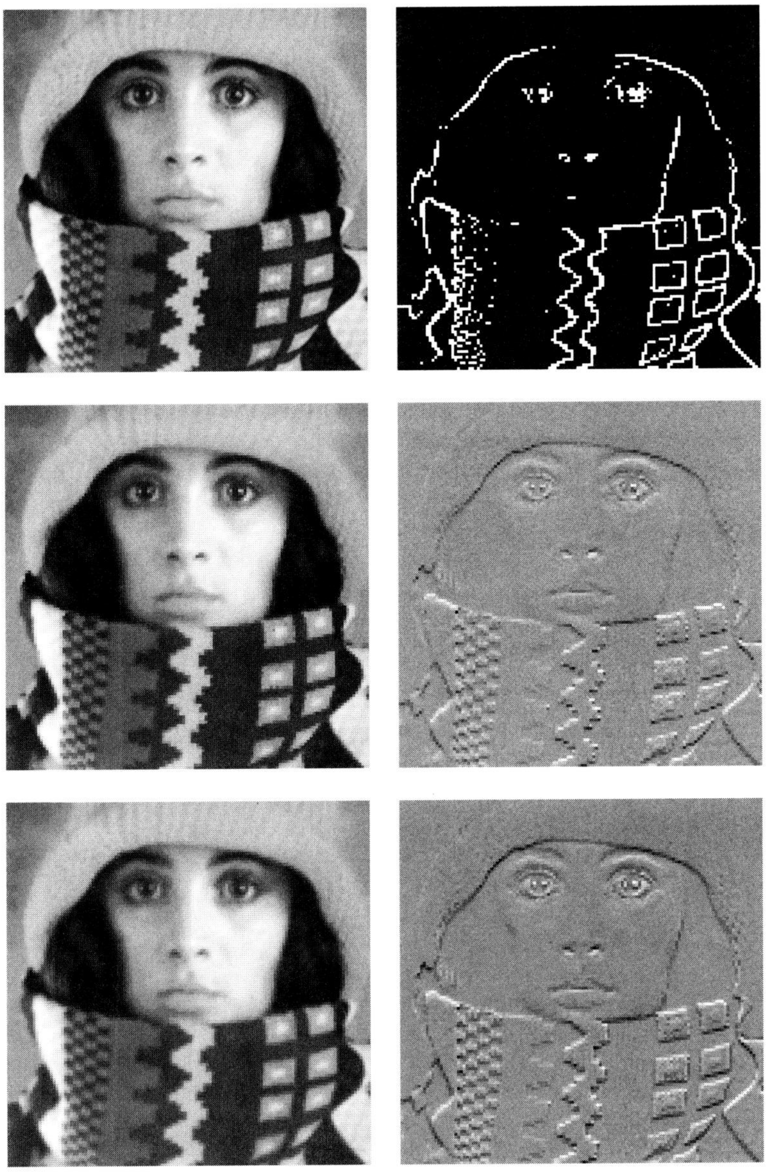


Figure 4.13: *Decompositions (at level* 1*) corresponding with Experiment 4.2.5. Top: original (left) and decision map (right) using a threshold* $T = 49$. *Middle: approximation (left) and horizontal detail (right) images in the adaptive case. Bottom: approximation (left) and horizontal detail (right) 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

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

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

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

$$|1 + \tfrac{\beta_{1,j}}{\alpha_1}| + (N-1)\tfrac{|\beta_{1,j}|}{|\alpha_1|} = \begin{cases} 1 + N\frac{|\beta_{1,j}|}{|\alpha_1|} > 1 & \text{if } \operatorname{sign}\beta_{1,j} = \operatorname{sign}\alpha_1 \\ 1 + (N-2)\frac{|\beta_{1,j}|}{|\alpha_1|} \geq 1 & \text{if } \operatorname{sign}\beta_{1,j} \neq \operatorname{sign}\alpha_1 \text{ and} \\ & \quad |\alpha_1| \geq |\beta_{1,j}| \\ N\frac{|\beta_{1,j}|}{|\alpha_1|} - 1 > 1 & \text{if } \operatorname{sign}\beta_{1,j} \neq \operatorname{sign}\alpha_1 \text{ and} \\ & \quad |\alpha_1| < |\beta_{1,j}|. \end{cases}$$

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 $N = 2$, $\operatorname{sign}\beta_{1,j} \neq \operatorname{sign}\alpha_1$ and $|\alpha_1| \geq |\beta_{1,j}|$. From these implications, Proposition 4.3.2 follows immediately. ☐

Next, we consider p to be the l^∞-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^∞-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

$$p_\infty(A_d^{-1}) = \max_i \left(|1 + \frac{\beta_{d,i}}{\alpha_d}| + \sum_{j \neq i} \frac{|\beta_{d,j}|}{|\alpha_d|} \right) .$$

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

$$p_\infty(A_0) = \begin{cases} 1 + \sum_j |\beta_{0,j}| > 1 & \text{if } \beta_{0,j} < 0 \text{ for some } j = 1, \dots, N \\ |1 - \beta_{0,m}| + \sum_{j \neq m} |\beta_{0,j}| & \begin{cases} = 1 & \text{if } N = 2 \text{ and } 0 \leq \beta_{0,1} = \beta_{0,2} \leq 1 \\ = 1 & \text{if } \beta_{0,j} = 0 \text{ for all } j = 1, \dots, N \\ > 1 & \text{otherwise}, \end{cases} \end{cases}$$

where $m = \operatorname{argmin}_j \beta_{0,j}$. Likewise, the l^∞-norm of A_1^{-1} is

$$p_\infty(A_1^{-1}) = \begin{cases} 1 + \sum_j \frac{|\beta_{1,j}|}{|\alpha_1|} > 1 & \text{if } \operatorname{sign} \alpha_1 = \operatorname{sign} \beta_{1,j} \text{ for some } j = 1, \dots, N \\ |1 + \frac{\beta_{1,m}}{\alpha_1}| + \sum_{j \neq m} \frac{|\beta_{1,j}|}{|\alpha_1|} & \begin{cases} = 1 & \text{if } N = 2,\ \beta_{1,1} = \beta_{1,2},\ \operatorname{sign} \alpha_1 \neq \operatorname{sign} \beta_{1,j} \\ & \text{and } |\beta_{1,j}| \leq |\alpha_1| \\ = 1 & \text{if } \beta_{1,j} = 0 \text{ for all } j = 1, \dots, N \\ > 1 & \text{otherwise}, \end{cases} \end{cases}$$

where $m = \operatorname{argmin}_j \frac{\beta_{1,j}}{\alpha_1}$. 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. □

4.3.2 Simulations

In this subsection we show some simulation results using the l^1-norm and the l^∞-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 $y'(n) = y(n) - \frac{1}{2}\big(x'(n) + x'(n+1)\big)$.

Experiment 4.3.1 (l^1-norm and l^∞-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

$$\begin{aligned} p_1(\boldsymbol{v}) &= |v_1| + |v_2| \\ p_\infty(\boldsymbol{v}) &= \max\{|v_1|, |v_2|\}. \end{aligned}$$

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^∞-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 $p_\infty(\boldsymbol{v}) \leq p_1(\boldsymbol{v})$, if $d = 1$ for the l^∞-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 $\boldsymbol{v} \in \mathbb{R}^N$, i.e.,

$$d = [p(\boldsymbol{v}) > T] \,.$$

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:

$$x' = \alpha_d x + \beta_{d,1} y_1 + \beta_{d,2} y_2 \tag{4.26}$$

where $d = D(\boldsymbol{v})$, 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 $\boldsymbol{v} = (v_1, v_2)^T$ 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 κ_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 $\boldsymbol{v}_d = (v_{d,1}, v_{d,2})^T$ 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

$$\begin{aligned} x' &= \alpha_a(\xi + v_{a,1}) + \beta_{a,1}\xi + \beta_{a,2}(\xi + c) \\ &= \kappa_a(\xi + v_{a,1}) - (\beta_{a,1} + \beta_{a,2})v_{a,1} + \beta_{a,2}c \\ &= \kappa_a\xi + \kappa_a v_{a,1} - \beta_{a,1}v_{a,1} - \beta_{a,2}v_{a,2} \,. \end{aligned}$$

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

$$x' = \kappa_b\xi + \kappa_b v_{b,1} - \beta_{b,1}v_{b,1} - \beta_{b,2}v_{b,2} \,.$$

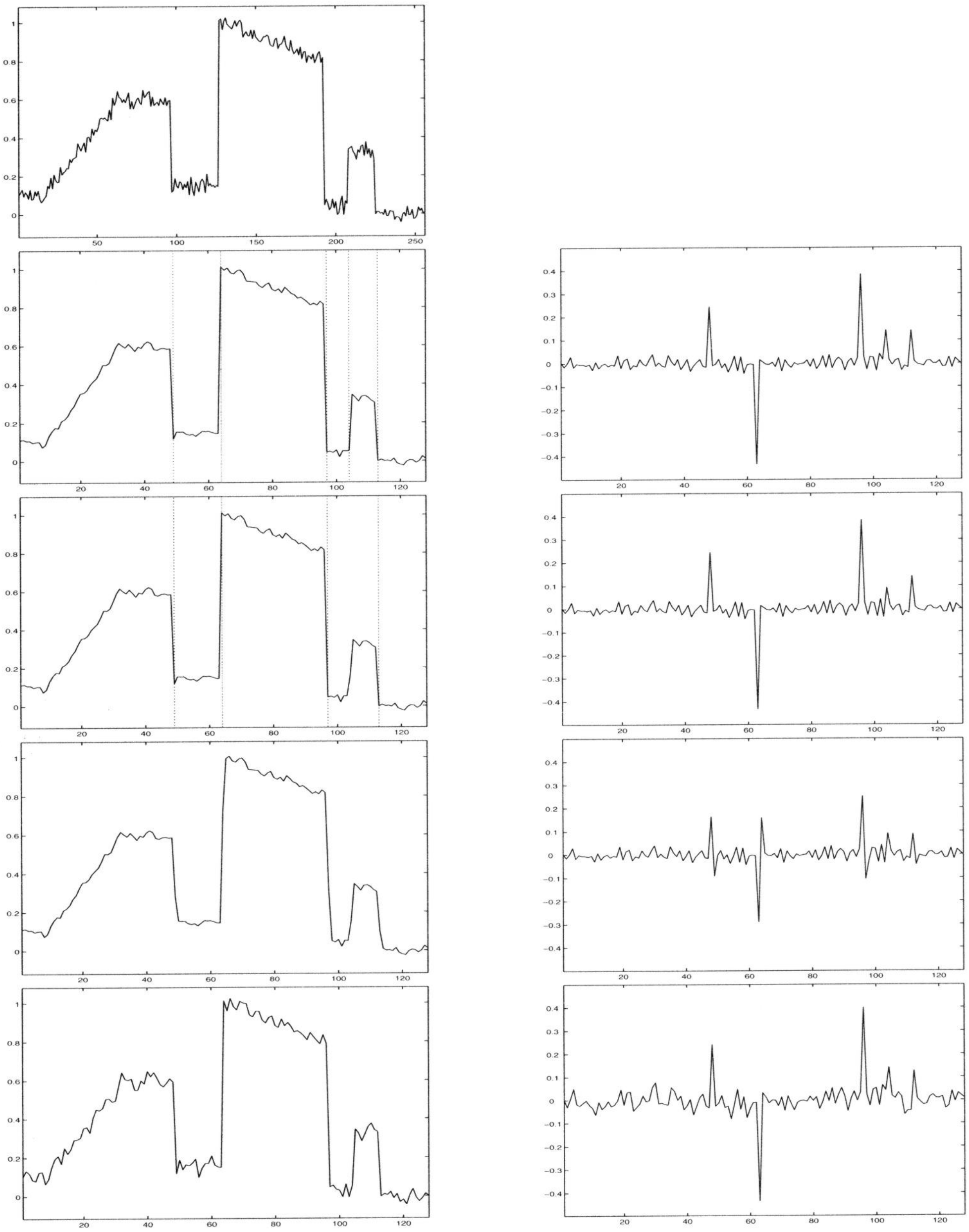


Figure 4.14: *Decompositions (at level* 1*) corresponding with Experiment 4.3.1. Top: original signal. Second and third rows: approximation (left) and detail (right) signals for the* l^1*-norm and the* l^∞*-norm, respectively, using a threshold* $T = 0.28$*. Fourth and bottom rows: approximation (left) and detail (right) signals in the non-adaptive cases with* $d = 0$ *and* $d = 1$*, respectively.*

Choose ξ in such a way that

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

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:

$$d = D\big(|v_1| + |v_2|\big) , \tag{4.27}$$

for D: $\mathbb{R}_+ \to \mathcal{D}$, 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 κ_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

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

Since A_d is invertible, we can recover $\boldsymbol{v}$ from $\boldsymbol{v}'$ 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 $\boldsymbol{v} = (v_1, v_2)^T \in \mathbb{R}^2$.

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

$$d = |x - y_1| + |x - y_2| = |v_1| + |v_2| , \tag{4.28}$$

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

$$x'_k = \alpha_{d_k}x_k + \beta_{d_k,1}y_1 + \beta_{d_k,2}y_2 ,$$

and whose corresponding gradient vector is $\boldsymbol{v}_k = (x_k - y_1, x_k - y_2)^T$. 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 $\boldsymbol{\beta}_{d_k}$ and the gradient components in $\boldsymbol{v}_k$:

$$\boldsymbol{\beta}_{d_k} = (\beta_{d_k}, \gamma_{d_k})^T \qquad \text{and} \qquad \boldsymbol{v}_k = (v_k, w_k)^T .$$

A straightforward computation shows that

$$\begin{aligned} x_2' - x_1' &= (\beta_{d_1} - \beta_{d_2})\Delta + \alpha_{d_1}(w_2 - w_1) + (\alpha_{d_2} - \alpha_{d_1})w_2 && (4.29) \\ &= (\gamma_{d_2} - \gamma_{d_1})\Delta + \alpha_{d_2}(v_2 - v_1) + (\alpha_{d_2} - \alpha_{d_1})v_1 , && (4.30) \end{aligned}$$

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 $x_2' - x_1' = \alpha_\Delta(w_2 - w_1)$. Since $w_2 - w_1 > 0$, we get that $\alpha_\Delta(w_2 - w_1) \neq 0$ has the same sign as α_Δ.
(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 β_d, γ_d are non-increasing, then $(\beta_{d_1} - \beta_{d_2})\Delta > 0$, $\alpha_{d_1}(w_2 - w_1) > 0$, and $(\alpha_{d_2} - \alpha_{d_1})w_2 > 0$. Hence, we get from (4.29) that $x_2' - x_1' > 0$. If $\alpha_d < 0$ and β_d, γ_d are non-decreasing, then all terms in (4.29) are negative, and we get $x_2' - x_1' < 0$.
(iii) $x_2 > y_2 > y_1 > x_1$: 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 $d_2 \geq d_1$, 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 β_d, γ_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 β_d, γ_d are non-decreasing, all terms in (4.30) are negative, and we get $x_2' - x_1' < 0$. □

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:*

$$\begin{aligned} x < y_1 &\iff x' < y_1 + \beta_{d,2}\Delta \\ y_1 \leq x \leq y_2 &\iff y_1 + \beta_{\Delta,2}\Delta \leq x' \leq y_2 - \beta_{\Delta,1}\Delta \\ x > y_2 &\iff x' > y_2 - \beta_{d,1}\Delta . \end{aligned}$$

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

$$\begin{aligned} x' &= x - \beta_{d,2}v_2 - \beta_{d,1}v_1 \\ &= y_2 + \alpha_d v_2 - \beta_{d,1}\Delta && (4.31) \\ &= y_1 + \alpha_d v_1 + \beta_{d,2}\Delta . && (4.32) \end{aligned}$$

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 $y_1 \le x \le y_2$, we have that $d = \Delta$, $v_1 \ge 0$, $v_2 \le 0$, and (4.31)-(4.32) yield that $y_1 + \beta_{\Delta,2}\Delta \le x' \le y_2 - \beta_{\Delta,1}\Delta$.

□

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 α_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 $y_1 \le y_2$. Observe that

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

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

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

and reconstruction becomes straightforward. If, however, $x' \notin [y_1 + \beta_{\Delta,2}\Delta, y_2 - \beta_{\Delta,1}\Delta]$, 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

$$\begin{aligned} Y &= \min\{y_1 + \beta_{\Delta,2}(y_2 - y_1), y_2 - \beta_{\Delta,1}(y_2 - y_1)\} \\ Z &= \max\{y_1 + \beta_{\Delta,2}(y_2 - y_1), y_2 - \beta_{\Delta,1}(y_2 - y_1)\}\,. \end{aligned}$$

(Note: the 'min' and 'max' are needed to cover both cases $y_1 \le y_2$ and $y_2 < y_1$.)

4. If $x' \in [Y, Z]$ (which implies $d = \Delta$) put
$$\beta_2 = \beta_{\Delta,2} \text{ and } \beta_1 = \beta_{\Delta,1} ;$$
otherwise

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

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

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
$$\beta_{d,1} = \beta_{d,2} = \frac{\beta_0}{\sigma d + 1} , \tag{4.33}$$
where $0 \le \beta_0 < \frac{1}{2}$ 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:
$$d = \frac{-(1 - 2\beta_0 - r\sigma) + \sqrt{(1 - 2\beta_0 - r\sigma)^2 + 4r\sigma}}{2\sigma} .$$
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
$$\beta_{d,j} = \begin{cases} \beta_{0,j} & \text{if } d \le T \\ \beta_{1,j} & \text{if } d > T \end{cases} \quad \text{for } j = 1, 2. \tag{4.34}$$
Proposition 4.4.3 yields the following result.

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.,

$$\begin{aligned} |v_1| + |v_2| > T \quad &\not\Rightarrow \quad |v_1'| + |v_2'| > T' \\ |v_1| + |v_2| \leq T \quad &\not\Rightarrow \quad |v_1'| + |v_2'| \leq T'\,. \end{aligned}$$

4.4.2 Simulations

We consider the case where the coefficients are given by (4.33) in Example 4.4.5, with $\beta_0 = \frac{1}{3}$ and $\sigma = 5$. That is,

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

Note that for small values of d, the resulting low-pass filter approximates the average filter ($\alpha = \beta_1 = \beta_2 = \frac{1}{3}$), 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 $\boldsymbol{v} \in \mathbb{R}^N$. In particular, we have studied the cases where the output of the decision map is given by:

(*a*) $d = [p(\boldsymbol{v}) > T]$ where p is a given seminorm and T is a threshold;

(*b*) $d = p(\boldsymbol{v})$ 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

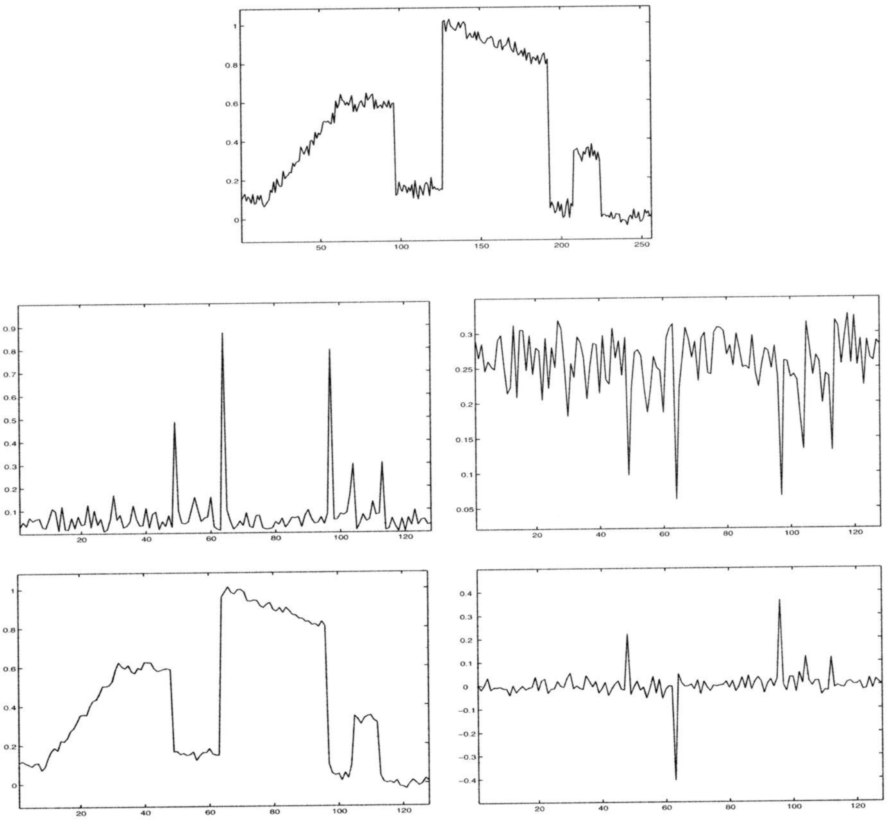


Figure 4.15: *Top: original signal. Middle: decision map d (left) and coefficients* $\beta_{d,1} = \beta_{d,2}$ *(right). Bottom: approximation (left) and detail (right) signals.*

such adaptive schemes. In all cases, as it will be shown, the resulting updated sample x' can be written as

$$x' = \alpha_d x + \sum_{j=1}^{N} \beta_{d,j} y_j \qquad \text{with } \kappa_d = \alpha_d + \sum_{j=1}^{N} \beta_{d,j} = 1 \text{ for all } d \in \mathcal{D}.$$

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,

$$x' = \alpha x + \beta y_s \,, \qquad \text{with } s = \operatorname{argmin}_j |x - y_j|. \tag{4.35}$$

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

$$\operatorname{argmin}_j |x - y_j| = \operatorname{argmin}_j |x' - y_j| \tag{4.36}$$

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

$$x = \frac{1}{\alpha}(x' - \beta y_s) \,, \qquad \text{with } s = \operatorname{argmin}_j |x' - y_j|. \tag{4.37}$$

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

$$|x' - y_j| = |(1 - \beta)x + \beta y_s - y_j| = |x - y_j - \beta(x - y_s)| \,. \tag{4.38}$$

If $y_j = y_s$, then $x - y_j = x - y_s$ 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 $x - y_j = y_s - x$ 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 $s = \operatorname{argmin}_j |x - y_j|$ 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

$$x' = \begin{cases} \frac{1}{2}x + \frac{1}{2}y_1 & \text{if } |v_1| < |v_2| \\ \frac{1}{2}x + \frac{1}{4}(y_1 + y_2) & \text{if } |v_1| = |v_2| \\ \frac{1}{2}x + \frac{1}{2}y_2 & \text{if } |v_1| > |v_2| \,, \end{cases}$$

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 $x' = \alpha_d x + \beta_{d,1} y_1 + \beta_{d,2} y_2$ with the three-valued decision map

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

where sign (0) = 0, and coefficients $\boldsymbol{\beta}_d$ given by

$$\boldsymbol{\beta}_{-1} = \tfrac{1}{2}(1,0)^T, \quad \boldsymbol{\beta}_0 = \tfrac{1}{4}(1,1)^T, \quad \boldsymbol{\beta}_{+1} = \tfrac{1}{2}(0,1)^T.$$

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 $|v_1'| = \frac{1}{2}|v_1|$ 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:

(*i*) $y_1 = y_2$: it is obvious that $|v_1'| = |v_2'|$, hence $d = 0$ at synthesis.

(*ii*) $v_1 = -v_2$: 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:

$$x' = \alpha x + \beta z, \qquad \text{with } z = \text{median}\{x, y_1, y_2, \dots, y_N\} \tag{4.39}$$

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

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

The proof is straightforward if we observe that

$$\text{median}\{x, y_1, y_2, \dots, y_N\} = \text{median}\{x', y_1, y_2, \dots, y_N\},$$

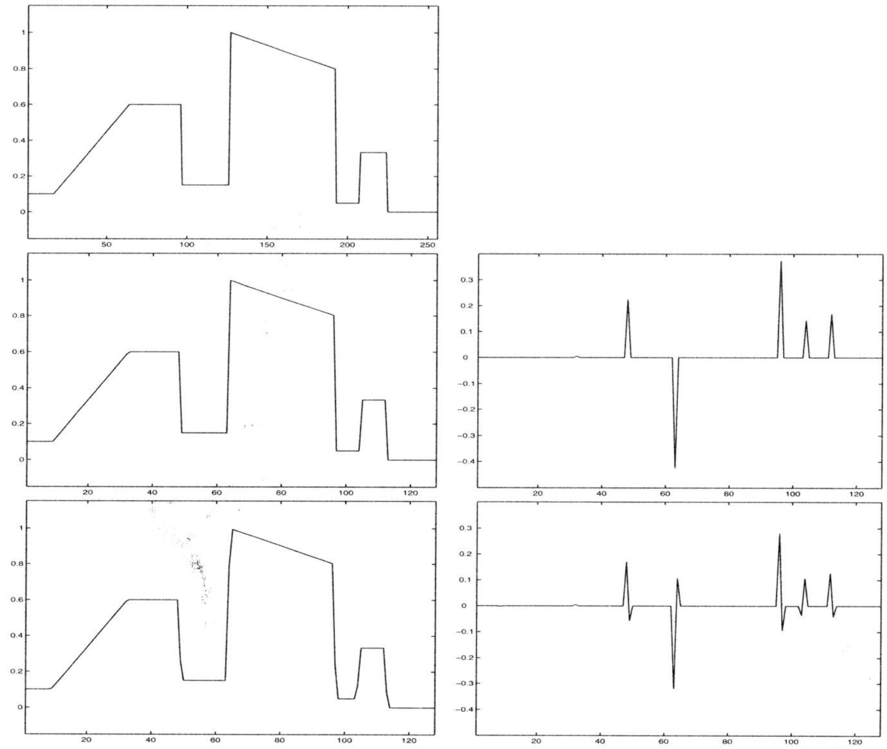


Figure 4.16: *Top: original signal. Middle: approximation (left) and detail (right) signals in the adaptive case. Bottom: approximation (left) and detail (right) signals in the non-adaptive case with* $d = 0$.

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

$$x' = \alpha_d x + \beta_d \mathrm{median}(x, y_1, y_2)\,, \tag{4.40}$$

where $d = [p(\boldsymbol{v}) > T]$, $\boldsymbol{v} \in \mathbb{R}^2$ and $p(\boldsymbol{v}) = \min\{|v_1|, |v_2|\}$. We choose $\boldsymbol{\beta}_1 = (0,0)^T$ and $\boldsymbol{\beta}_0 = \frac{1}{2}(1,1)^T$. Then we can write (4.40) as

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

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

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

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

$$d = [p_h(\boldsymbol{v}) \leq p_v(\boldsymbol{v})] . \tag{4.41}$$

The filter coefficients $\boldsymbol{\beta}_d$ are chosen as follows:

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

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 $p_h(\boldsymbol{v}) \leq p_v(\boldsymbol{v})$, 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

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

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

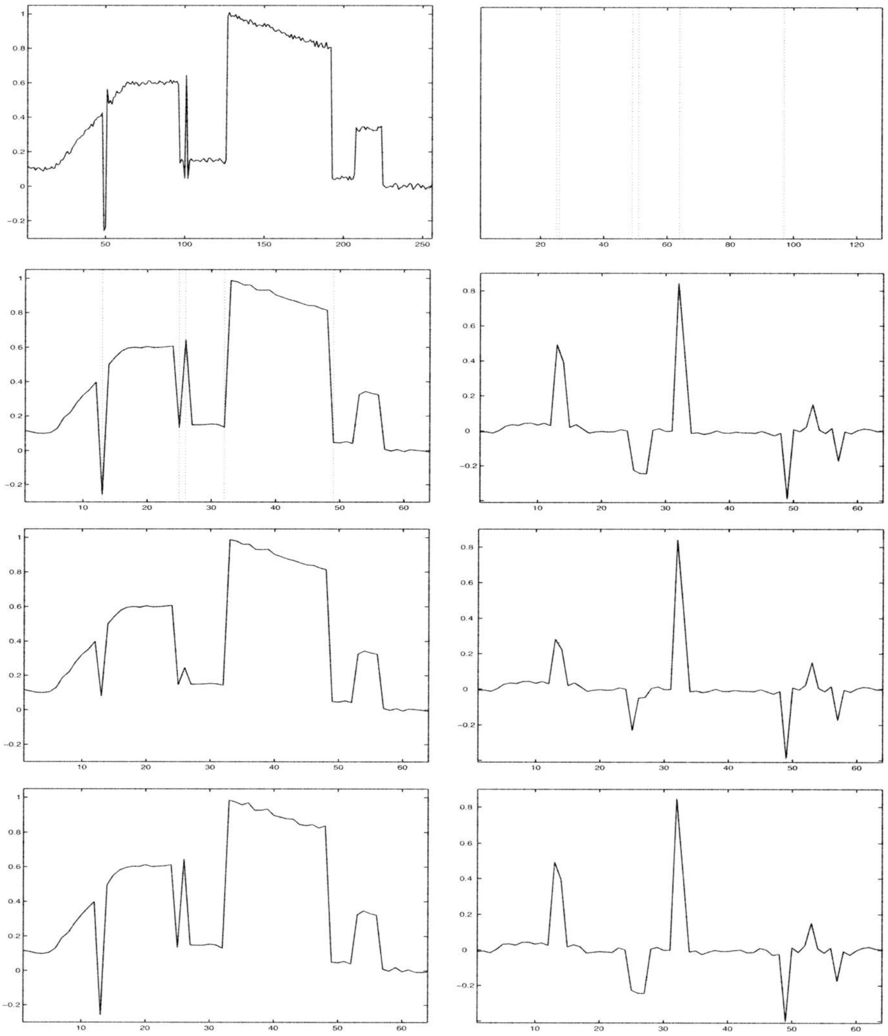


Figure 4.17: *Decompositions (at level* 2*) corresponding with Experiment 4.5.2. Top: original signal (left) and decision map at level* 1 *(right). Second row: approximation (left) and detail (right) signals in the adaptive case. Third and bottom rows: approximation (left) and detail (right) signals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

	$y_2(\boldsymbol{n})$	
$y_3(\boldsymbol{n})$	$x(\boldsymbol{n})$	$y_1(\boldsymbol{n})$
	$y_4(\boldsymbol{n})$	

Figure 4.18: *Indexing of samples in a* 3×3 *window centered at* $x(\boldsymbol{n})$.

Proposition 4.5.1. *To have perfect reconstruction it is sufficient that*

$$
\begin{aligned}
0 \le \beta_0 < \frac{1}{4} \le \gamma_0 < \frac{1}{2} \quad &\textit{and} \quad \beta_0 + \gamma_0 < \frac{1}{2} \\
0 \le \gamma_1 < \frac{1}{4} \le \beta_1 < \frac{1}{2} \quad &\textit{and} \quad \beta_1 + \gamma_1 < \frac{1}{2},
\end{aligned}
$$

and in this case the decision d *can be recovered at synthesis from* $d = [p_h(\boldsymbol{v}') \le p_v(\boldsymbol{v}')]$.

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

$$H = v_1 + v_3 \text{ and } V = v_2 + v_4\,,$$

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

$$
\begin{aligned}
H' &= (1 - 2\beta_d)H - 2\gamma_d V &\quad (4.43)\\
V' &= -2\beta_d H + (1 - 2\gamma_d)V\,. &\quad (4.44)
\end{aligned}
$$

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

(*i*) $0 \le H < V$: then

$$
\begin{aligned}
0 < V' &= 2\beta_1(V - H) + (1 - 2\gamma_1 - 2\beta_1)V \\
H' &= -(1 - 2\beta_1)(V - H) + (1 - 2\gamma_1 - 2\beta_1)V\,.
\end{aligned}
$$

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

$$0 < (1 - 2\beta_1)(V - H) \le 2\beta_1(V - H)\,,$$

it follows that $|H'| \le |V'|$.

(*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 $1 - 2\beta_1 < 2\beta_1$ 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

$$\begin{aligned} H' &= (1 - 2\beta_1)(H - V) + (1 - 2\beta_1 - 2\gamma_1)V \\ V' &= -2\beta_1(H - V) + (1 - 2\beta_1 - 2\gamma_1)V\,. \end{aligned}$$

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

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.

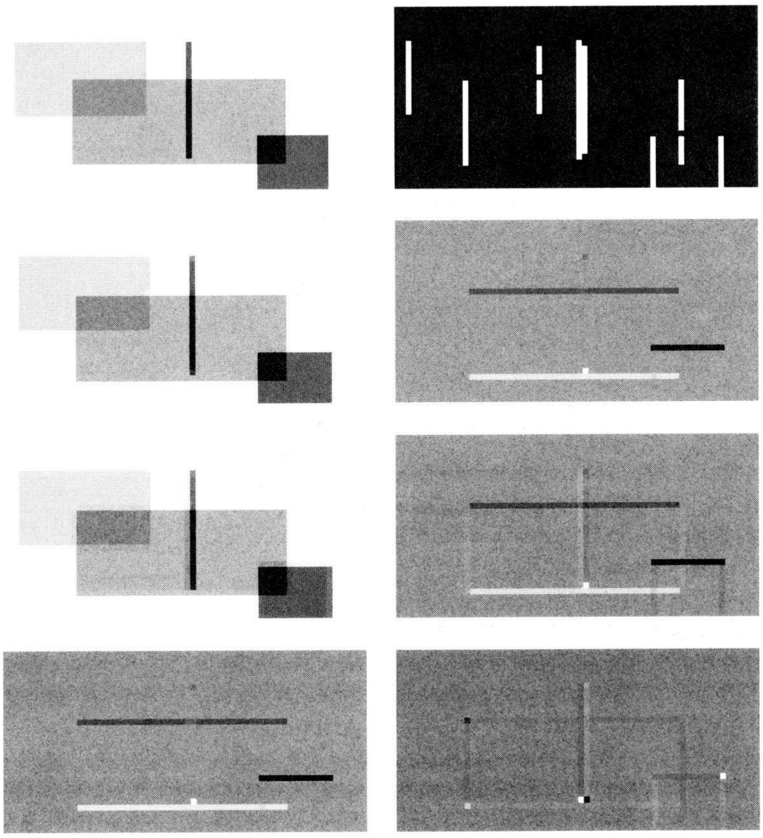


Figure 4.19: *Decompositions (at level 1) corresponding with Experiment 4.5.3. Top: input image (left) and decision map (right). Second row: approximation (left) and horizontal detail (right) images in the adaptive case. Third row: approximation (left) and horizontal detail (right) images in the non-adaptive case with d = 0. Bottom: diagonal detail images in the adaptive (left) and non-adaptive (right) cases.*

Chapter 5

Adaptive wavelets in image compression

The development of image compression methods is an on-going theoretical and practical research effort. Video telephony, teleconferencing, accessing images from distant servers, video communications as well as many other multimedia applications would not be feasible without compression. In addition, most of these applications request new functionalities such as object-oriented coding and progressive transmission of information.

For various reasons wavelet-based image compression algorithms are becoming extremely popular, and they have been adopted into the new still-image coding standard JPEG2000. Such compression algorithms exploit the ability of wavelet representations to efficiently decorrelate and approximate image data with few non-zero wavelet coefficients. Moreover, the multiresolution nature of wavelets allows for progressive image coding, a useful functionality for image transmission over low bandwidth channels.

In this chapter, we examine the potential of the adaptive wavelet schemes described in Chapter 3 for image compression. Section 5.1 starts with a reminder of basic concepts from image compression. In Section 5.2, we discuss briefly wavelet-based image compression. In Section 5.3, we point out some limitations of classical wavelets. We mention various new approaches that have been recently introduced to overcome such limitations and outline the adaptive wavelet schemes proposed in previous chapters. In Section 5.4, we evaluate the effectiveness of our adaptive wavelet schemes by means of an entropy criterion. In Section 5.5, we discuss the effects of quantization in the adaptive wavelet scheme. We provide conditions for recovering the original decisions at synthesis and we provide expressions that relate the reconstruction error to the quantization error. Such an analysis is essential for the application of our adaptive decompositions in lossy image compression algorithms. In Section 5.6 we show several simulation results. Finally, in Section 5.7, we give some conclusions and discuss future research.

5.1 Preliminaries

Compression is a process intended to yield a compact representation of the data by removing, or at least reducing, the redundancy present in the original data. For example, in images there is usually a considerable amount of correlation among nearby pixels. Thus, a major ingredient of an image compression method is the design of an image representation that captures this redundancy and hence reduces the number of bits required to represent or approximate the image. In practice, this amounts to a constrained optimization procedure that tries to minimize the number of bits while maintaining a certain quality.

The degree of compression is usually measured by the so-called *compression ratio.* This is defined as the ratio of the number of bits required to represent the image before compression over the number of bits required to represent it after compression. The average number of bits used to represent each sample value (i.e, pixel) is referred to as the *bit rate* and is usually expressed as *bpp* (bits per pixel).

There are two basic kinds of compression schemes: lossless and lossy. Lossless schemes compress the image without loss of information so that the original image can be recovered exactly from its compressed version. Lossy compression schemes involve some loss of information. After lossy compression, the original image cannot be perfectly reconstructed. In return for accepting an error in the reconstruction, one can achieve higher compression ratios than with lossless compression.

In lossy compression, the reconstruction differs from the original signal. The difference between the original and reconstructed signal is referred to as approximation error or *distortion.* Although there are several approaches to measure such distortion, the most commonly used are the *mean squared error* (MSE) and the *peak-signal-to-noise-ratio* (PSNR). They are respectively defined as:

$$\mathrm{MSE} = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\big(x(m,n)-\hat{x}(m,n)\big)^2 , \tag{5.1}$$

where x is the original image of size $M \times N$ and $\hat{x}$ is the reconstructed image, and

$$\mathrm{PSNR} = 10\log_{10}\frac{(x_{\mathrm{max}})^2}{\mathrm{MSE}} , \tag{5.2}$$

where x_{max} is the maximum possible intensity value in the image (e.g., $x_{\mathrm{max}} = 2^b - 1$ for images with b bits depth). Obviously, smaller MSE and larger PSNR values correspond to lower levels of distortion[1]. However, it is important to note that these distortion measures do not always correlate well with image quality as perceived by the human visual system. This is particularly true at high compression ratios (i.e., low bit rates).

In practice, the distortion increases with the amount of compression, that is, distortion is a function of (bit) rate. For this reason, plots of distortion D versus rate R are often used to analyze lossy compression performance. Rate-distortion theory [37, 127, 150] is concerned with the trade-offs between distortion and rate in lossy compression schemes. One way of representing the trade-offs is via a rate-distortion function $R(D)$, which specifies the lowest rate at which the signal can be encoded while keeping the distortion less or equal to D.

In lossless compression, there is no distortion but there is only a limited amount of compression that can be obtained, depending on the information content of the image. Intuitively,

---

[1] Other terms related to distortion are *fidelity* and *quality.* When we say that the fidelity or quality of a reconstructed image is high, we mean that the distortion is small.

the amount of information conveyed by the data depends upon its 'randomness'. The more randomness, the less redundancy there is among the data and hence the more difficult it is to compress it. An extreme case is white noise which is incompressible. The coding efficiency in lossless schemes is sometimes characterized by the *entropy* of the compressed representation of the image.

The entropy of a discrete random variable X is defined as

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x) , \tag{5.3}$$

where $\mathcal{X}$ is the set of possible outcomes of X, and $p(x)$ is the probability of the outcome $X = x$. The entropy is a measure of the amount of information required on the average to describe the random variable. It is an important concept for understanding and developing compression algorithms: Shannon's coding theorem [132, 133] says that the entropy of X represents a lower bound on the average number of bits needed to represent each of its outcomes.

A typical compression algorithm consists of three steps: transformation, quantization and coding. The *transformation* step applies a transform to the original image so that the resulting set of coefficients are more amenable to compression. The transforms are designed to remove the statistical correlation and/or to separate irrelevant (e.g., from a perceptual point of view) information from relevant information. The transform operation is usually invertible. *Quantization* is used to discard transform coefficient information that is considered to be 'insignificant'. Quantization maps the output set of the transform into a smaller set. In most cases, it is only the quantization step that discards information and hence introduces distortion. In the case of lossless compression, no quantization is performed. The *coding* process exploits the statistical redundancy in the (quantized or not) transform coefficients. This step is invertible.

The decompression algorithm simply mirrors the process used for compression. First, the compressed image is decoded to obtain the quantized transform coefficients. These are then 'de-quantized' yielding an approximation of the original transform coefficients. Finally, the inverse of the transform used during compression is employed to obtain a reconstructed image.

The kind of compression schemes described above are also known as *transform-based compression schemes*. Note that the transform step is only one of the three components of such compression steps, and that there exists a strong interplay among them. In this thesis, however, we are mostly interested in the transform step, and more specifically, in the adaptive wavelet transform schemes described in previous chapters. We briefly address quantization, but the coding step is beyond the scope of the thesis.

5.2 Wavelets in image compression

One popular choice for the transformation step is the wavelet transform, which has proved quite effective for image compression [15, 47, 61, 90, 125, 134]. The effectiveness of wavelets stems from the fact that, thanks to their good space-frequency localization, they can provide a sparse representation of images with significant wavelet coefficients occurring mostly in the neighborhood of edges and other kind of strong transitions. The intuitive explanation is that images typically consist of smooth areas separated by sharp transitions (i.e., edges). Within the smooth regions, wavelet coefficients are small (due to vanishing moments and regularity properties) and they decay rapidly from coarse to fine scales. In the neighborhood of edges, wavelet coefficients are larger and they decay in a slower fashion, but because of their local support, relatively few wavelet coefficients are affected by the presence of edges. Thus, wavelet decompositions result in few large-amplitude coefficients which, in addition, correspond to visually important image features (i.e, edges).

Another reason to take recourse to wavelets is their multiresolution nature, which facilitates functionalities such as progressive transmission and resolution scalability (i.e., a compressed bit-stream may be partially decompressed to obtain successively higher resolution versions of the original image).

In addition, wavelet decompositions exhibit dependencies across and within scales which can be easily exploited by the quantization and coding steps. For example, if a wavelet coefficient is small, then it is likely that its descendants, that is, coefficients corresponding to the same spatial location in finer scales, are also small. In such case one may code such wavelet coefficient and all its descendants into a single 'zero' symbol. This is the idea behind the so-called wavelet *zero-tree coders*. A well-known example is the embedded zero-tree wavelet (EZW) algorithm developed by Shapiro [134], which combines the zero-tree coding with bit-plane coding. 'Embedded' means that the coder can stop encoding at any desired rate (progressive transmission). In the EZW algorithm this is done by transmitting more important information first.

5.3 Adaptive wavelets in image compression

A key point in the success of wavelets for compression is their good nonlinear approximation properties for piecewise smooth functions in one dimension (1D) [47]. Unfortunately, this is less true for two-dimensional (2D) functions. Wavelets in 2D are usually obtained by a tensor product of 1D wavelets. Thus, they are adapted to point singularities and cannot efficiently model higher order singularities, like edges in images. To a large extent the same can be said for other basic constructions of non-separable 2D wavelets. This intrinsic weakness of classical 2D wavelets has motivated various researchers to look for novel wavelet representations that are better suited for the description of images. Several promising approaches are currently under investigation, including compression in the ridgelet and curvelet domain [48], compression along curves using bandelets [88], and edgeprints [53]. In Section 3.1, some of these approaches have been discussed in more detail.

In the same venue, the adaptive wavelet schemes that we have described in previous chapters may be regarded as another attempt to overcome the limitations of classical wavelets. In the following sections, we investigate the potential of our adaptive schemes in image compression. We emphasize that the results reported here concern only a very first research effort and that much more investigation will be required to get a good understanding of the potentials of our adaptive wavelets schemes.

We restrict ourselves to the adaptive wavelet schemes described in Section 3.4. That is, an input signal x^0: $\mathbb{Z}^d \to \mathbb{R}$ is split into x, $y = \{y(\cdot|1), \ldots, y(\cdot|P)\}$. These bands pass through a two-stage lifting system comprising an adaptive update lifting step which returns

$$x'(\boldsymbol{n}) = \alpha_{d_{\boldsymbol{n}}} x(\boldsymbol{n}) + \sum_{j=1}^{N} \beta_{d_{\boldsymbol{n}},j} y_j(\boldsymbol{n}) \,, \tag{5.4}$$

with $y_j(\boldsymbol{n}) = y(\boldsymbol{n} + \boldsymbol{l}_j | p_j)$, $p_j \in \{1, \ldots, P\}$, and a fixed prediction step yielding

$$y'_j(\boldsymbol{n}) = y_j(\boldsymbol{n}) - P_j(x', y')(\boldsymbol{n}) \,. \tag{5.5}$$

The update step is triggered by the outcome $d_{\boldsymbol{n}} = D\big(\boldsymbol{v}(\boldsymbol{n})\big) \in \{0, 1\}$ of the decision map D. We assume that

$$D(\boldsymbol{v}) = [p(\boldsymbol{v}) > T] \,,$$

where p is a seminorm, T is a threshold and $\boldsymbol{v} \in \mathbb{R}^N$ is the gradient vector with components v_j given by

$$v_j(\boldsymbol{n}) = x(\boldsymbol{n}) - y_j(\boldsymbol{n}) \,, \quad j = 1, \ldots, N \,.$$

For the coefficients in (5.4) we assume that

$$\alpha_d + \sum_{j=1}^{N} \beta_{d,j} = 1 \quad \text{for } d = 0, 1 \,.$$

We have seen in Section 3.4 that the gradient vector $\boldsymbol{v}' \in \mathbb{R}^N$ with components $v'_j(\boldsymbol{n}) = x'(\boldsymbol{n}) - y_j(\boldsymbol{n})$, $j = 1, \ldots, N$, is related to $\boldsymbol{v}$ by means of the linear relation

$$\boldsymbol{v}' = A_d \boldsymbol{v} \,, \tag{5.6}$$

where $A_d = I - \boldsymbol{u}\boldsymbol{\beta}_d^T$ with $\boldsymbol{u} = (1, \ldots, 1)^T$ and $\boldsymbol{\beta}_d = (\beta_{d,1}, \ldots, \beta_{d,N})^T$.

It is obvious that the inversion of (5.4) is straightforward if $d_{\boldsymbol{n}}$ is known. We have shown in Section 3.4 that if the threshold criterion holds (i.e, $p(A_0)$ and $p^{-1}(A_1)$ are finite and $p(A_0)p^{-1}(A_1) \leq 1$), it is possible to recover the decision at analysis which is based on the gradient vector $\boldsymbol{v}$, from the gradient $\boldsymbol{v}'$ which is available at synthesis.

The combination of the adaptive update lifting with the fixed prediction lifting yields an adaptive wavelet decomposition step mapping x^0 into x', y'. We obtain an adaptive multiresolution wavelet decomposition by iteration of such wavelet steps. That is, we can use the approximation x' as the input for another wavelet decomposition step and obtain x'', y''. Then, we can repeat this for x'', and so on. Thus, iteration of K steps results in a K-level wavelet decomposition of x^0 into $y^1, y^2, \ldots, y^K, x^K$, where we have written y^1, y^2, etc, instead of y', y'', etc., for simplicity of notation.

5.4 Computing the empirical entropy

In this section we investigate the potential of our adaptive wavelet schemes to yield an effective representation for compression purposes. In particular, we evaluate the coding efficiency of some of the decompositions proposed in Chapter 4 by using a measure that is directly related to simple rate compression algorithms.

In Section 5.1 we introduced the concept of entropy for a discrete random variable X. Consider now an image $x\colon \mathbb{Z}^2 \to \mathbb{Z}$ of size $M \times N$, whose pixels can take L distinct values, say $0, 1, \ldots, L-1$. We define the *empirical distribution* p_x of x by

$$p_x(l) = \frac{1}{MN}\mathrm{card}\{\boldsymbol{n} \in \mathbb{Z}^2 \mid x(\boldsymbol{n}) = l\}\,, \qquad l = 0, 1, \ldots, L-1\,,$$

i.e., p_x is the L-bin normalized histogram of x. Analogous to (5.3), we define the *empirical entropy*:

$$H(x) = -\sum_{l=0}^{L-1} p_x(l)\log_2 p_x(l)\,. \tag{5.7}$$

Our measure takes into account the fact that the statistics in different bands of a wavelet transform are different. For each band, we first uniformly quantize the transform coefficients with 256 bins. Then, we calculate the empirical entropy as in (5.7), where x is the 256-bin quantized band.

The overall entropy (in bits per pixel) of a K-level decomposition is then calculated as the weighted sum of the (empirical) entropies of the approximation x^K and the details $y^1, \ldots, y^K$ bands:

$$h = 2^{-2K}H(x^K) + \sum_{k=1}^{K}\sum_{p=1}^{3} 2^{-2k}H\big(y^k(\cdot|p)\big)\,, \tag{5.8}$$

for the 2D square sampling, and

$$h = 2^{-K}H(x^K) + \sum_{k=1}^{K} 2^{-k}H(y^k)\,, \tag{5.9}$$

for the quincunx sampling.

In our simulations, we use the images shown in Fig. 5.1-Fig. 5.2 and apply some of the adaptive decompositions used in the experiments[2] of Chapter 4. We compute the empirical entropy of each original image and the overall empirical entropy of each decomposition as described above. In the sequel, we use simply the term 'entropy', since it is clear from the context which definition ((5.7), (5.8) or (5.9)) is being used.

Tables 5.1-5.2 show the entropy values for $K = 2$ and $K = 4$ levels of decomposition, respectively, for the images in Fig. 5.1. In each case, the entropy value of the corresponding non-adaptive decomposition with fixed $d = 0$ is also given. The entropy of the original images is shown in the second rows. From these tables, one can see that in all cases the adaptive decompositions give lower entropies than their non-adaptive counterparts, and that the 4-level decompositions (Table 5.2) provide a more compact representation (lower entropy) than the 2-level decompositions (Table 5.1). Moreover, for $K = 4$, the coding efficiency of the adaptive

---

[2] We take the same filters and decision map for these experiments, except for the threshold value T, which we choose in such a way that 15% of the decision map at the first level has value 1.

decompositions is considerably higher than that of the non-adaptive case, whereas a smaller improvement is observed for $K = 2$.

We also observe that, in terms of the proposed entropy measure and for the chosen set of images, the adaptive transform in Experiment 4.2.5 performs the best, followed by the adaptive transform in Experiment 4.1.4. Experiment 4.2.5 corresponds to a square sampling decomposition using a weighted quadratic seminorm. Experiment 4.1.4 corresponds also to a square sampling decomposition but using a weighted gradient seminorm. In both cases, the equivalent low-pass analysis filter is a 3×3 low-pass filter for $d = 0$ and the identity filter for $d = 1$.

Figure 5.1: *Test images. From left to right and from top to bottom: 'House', 'Lenna', 'Barbara' and 'Goldhill'. Their respective sizes are:* 256×256, 512×512, 576×704 *and* 576×704.

Table 5.3 shows the entropy values for $K = 2$ taking as input images the synthetic images depicted in Fig. 5.2. We evaluate the coding performance of Experiment 4.2.5 and Experiment 4.5.3. In this latter experiment, the adaptive scheme switches between horizontal and vertical low-pass filters of length 3. The non-adaptive case corresponds to a filter which averages each sample with the average of its four horizontal and vertical neighbors.

In both experiments the entropy values of the adaptive scheme are considerably smaller than for their non-adaptive counterparts. This is to be expected since the images comprise large homogeneous regions separated by sharp edges, and our adaptive decompositions are especially suited to distinguish between 'edge' and 'non-edge' regions. Note that for image

Experiment		House	Barbara	Goldhill	Lenna
		6.23	7.54	7.56	7.44
4.1.3	adaptive	**5.64**	**6.00**	**5.95**	**5.85**
	$d = 0$	5.77	6.15	6.03	5.86
4.1.4	adaptive	**5.01**	**5.45**	**5.41**	**5.20**
	$d = 0$	5.09	5.48	5.47	5.30
4.2.3	adaptive	**5.53**	**5.93**	**5.88**	**5.75**
	$d = 0$	5.77	6.15	6.03	5.86
4.2.4	adaptive	**5.51**	**5.97**	**5.98**	**5.79**
	$d = 0$	5.80	6.27	6.08	5.96
4.2.5	adaptive	**4.84**	**5.42**	**5.31**	**5.12**
	$d = 0$	5.30	5.89	5.75	5.36

Table 5.1: *Entropy values for the adaptive and non-adaptive ($d = 0$) decomposition schemes using* $K = 2$.

Experiment		House	Barbara	Goldhill	Lenna
		6.23	7.54	7.56	7.44
4.1.3	adaptive	**5.35**	**5.67**	**5.51**	**5.35**
	$d = 0$	5.54	5.86	5.70	5.45
4.1.4	adaptive	**4.90**	**5.35**	**5.29**	**5.06**
	$d = 0$	5.00	5.39	5.35	5.18
4.2.3	adaptive	**5.11**	**5.53**	**5.34**	**5.16**
	$d = 0$	5.54	5.86	5.70	5.45
4.2.4	adaptive	**5.08**	**5.58**	**5.44**	**5.23**
	$d = 0$	5.60	5.97	5.78	5.62
4.2.5	adaptive	**4.70**	**5.31**	**5.17**	**4.96**
	$d = 0$	5.22	5.81	5.66	5.25

Table 5.2: *Entropy values for the adaptive and non-adaptive ($d = 0$) decomposition schemes using* $K = 4$.

'Synth2', which contains diagonal edges, the improvements over the non-adaptive approaches are smaller than for image 'Synth1', which contains only horizontal and vertical edges. This is most noticeable in Experiment 4.5.3, where the filtering does not take into account the diagonal neighbors.

5.5 Adding quantization

Let $x^k \colon \mathbb{Z}^d \to \mathbb{R}$ denote the various levels of the signal for $k = 0, 1, \ldots, K$. Given x^{k-1} we obtain x^k, y^k by applying some wavelet transform W. The output approximation signal x^k is used as input for the next block and the output detail signal y^k is quantized by a mapping Q yielding

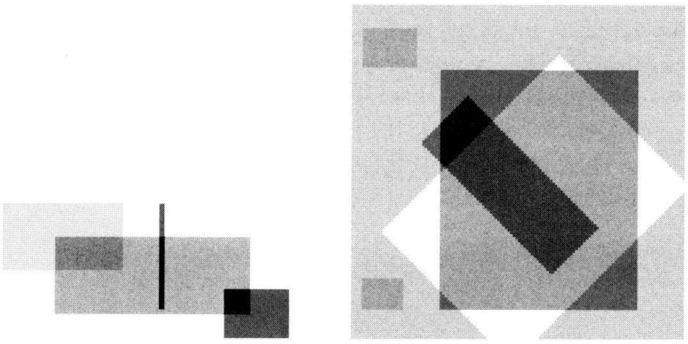


Figure 5.2: *Synthetic test images: 'Synth1' (left) and 'Synth2' (right). Their respective sizes are* 64×128 *and* 128×128.

Experiment		Synth1	Synth2
		1.79	2.01
4.2.5	adaptive	**0.40**	**0.62**
	$d = 0$	1.33	1.63
4.5.3	adaptive	**0.40**	**0.77**
	non-adapt.	1.27	1.24

Table 5.3: *Entropy values for the adaptive and non-adaptive decomposition schemes using* $K = 2$.

$\tilde{y}^k = Q(y^k)$. After a K-level decomposition we have K quantized detail signals $\tilde{y}^1, \ldots, \tilde{y}^K$ and an approximation signal x^K which is also quantized by Q to yield $\tilde{x}^K$. This decomposition scheme is shown in Fig. 5.3, where the wavelet transform W corresponds to the adaptive lifting scheme described in Chapter 3 and summarized in Section 5.3. Here, however, we assume that the prediction is based only on the approximation signal. The transform W is also illustrated in Fig. 5.3. Here, S is an invertible mapping which splits x^{k-1} into two components x_0^k, y_0^k, and Γ is the gradient operator, e.g., $\boldsymbol{v}_0^k = \Gamma(x_0^k, y_0^k)$.

The synthesis scheme that we use is depicted in Fig. 5.4. Here U^{-1} and S^{-1} denote the inverse of the update lifting and the splitting, and Q^{-1} is a right inverse of Q, i.e., $QQ^{-1}(t) = t$ for $t \in \operatorname{Ran}(Q)$. We assume that for all $z \in \mathbb{R}$,

$$|Q^{-1}Q(z) - z| \leq \nu q\,,$$

where q is the *quantization parameter.* For example, if $Q(z) = \{z/q\}$, where $\{\cdot\}$ denotes rounding to the closest integer, then $\nu = 1/2$. Henceforth we shall restrict ourselves to this particular case, which is known in the literature as *uniform quantization.* Moreover, we shall use the same quantizer Q for all bands.

We assume that the threshold criterion holds. In the scheme of Fig. 5.3 this means that we will be able to recover $d_{\boldsymbol{n}}^k = D(\boldsymbol{v}_0^k(\boldsymbol{n}))$, where $\boldsymbol{v}_0^k(\boldsymbol{n}) = \Gamma(x_0^k, y_0^k)(\boldsymbol{n})$, from $\boldsymbol{v}^k(\boldsymbol{n}) = \Gamma(x^k, y_0^k)(\boldsymbol{n})$, where x^k is the update of x_0^k. However, as becomes clear from the scheme in Fig. 5.4, we can

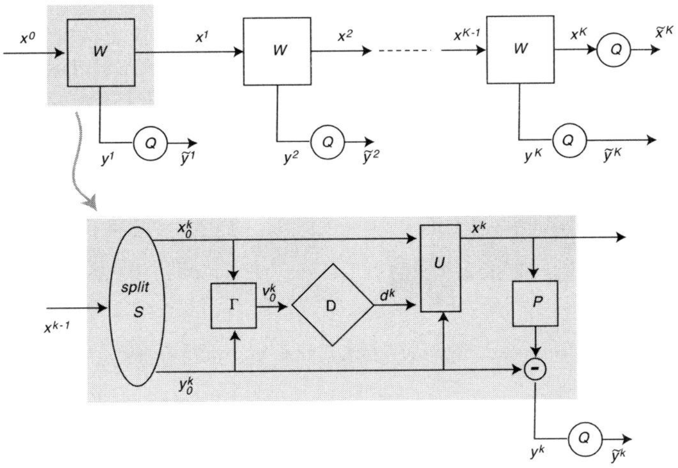


Figure 5.3: *Analysis part of a multiresolution wavelet decomposition scheme with quantization. The wavelet transform W (gray box) is obtained by a splitting S, an adaptive update step U and a fixed prediction step P.*

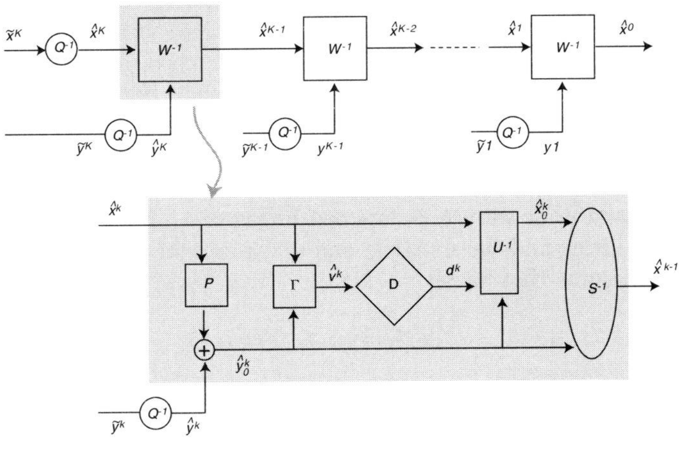


Figure 5.4: *Synthesis part of a multiresolution wavelet decomposition scheme with quantization.*

only dispose of $\hat{\boldsymbol{v}}^k(\boldsymbol{n}) = \Gamma(\hat{x}^k, \hat{y}_0^k)(\boldsymbol{n})$, where $\hat{x}^k$, $\hat{y}_0^k$ are 'approximations' of x^k, y_0^k due to errors resulting from the various quantization steps. Obviously, since the quantization step discards information, the synthesis scheme does not invert the analysis part. Below we derive estimates for the difference between the original signal x^0 and the synthesized signal $\hat{x}^0$. In particular, we derive conditions that guarantee that we can recover each original decision $d_{\boldsymbol{n}}^k$ at synthesis. This is important, since 'wrong' decisions at synthesis inevitably lead to 'bad' reconstructions. Before we state the precise result, we give some notation.

In the sequel, we denote by $\boldsymbol{y}(\boldsymbol{n})$ the vector $\big(y_1(\boldsymbol{n}), \ldots, y_N(\boldsymbol{n})\big)^T$. For a vector $\boldsymbol{y} \in \mathbb{R}^N$, the notation $|\boldsymbol{y}|$ denotes the l^∞-norm, i.e., $|\boldsymbol{y}| = \max\{|y_1|, \ldots, |y_N|\}$. If x: $\mathbb{Z}^d \to \mathbb{R}$ and $\boldsymbol{y}$: $\mathbb{Z}^d \to \mathbb{R}^N$, then $\|\cdot\|$ denotes the sup-norm, i.e.,

$$\|x\| = \sup\{|x(\boldsymbol{n})| : \boldsymbol{n} \in \mathbb{Z}^d\}, \quad \|\boldsymbol{y}\| = \sup\{|\boldsymbol{y}(\boldsymbol{n})| : \boldsymbol{n} \in \mathbb{Z}^d\}.$$

Proposition 5.5.1. *Assume that $\rho, \pi_x, \pi_y > 0$ are constants such that*

$$|P(x)(\boldsymbol{n}) - P(x')(\boldsymbol{n})| \le \rho\|x - x'\|, \tag{5.10}$$

$$p\big(\boldsymbol{v}(\boldsymbol{n}) - \boldsymbol{v}'(\boldsymbol{n})\big) \le \pi_x\|x - x'\| + \pi_y\|\boldsymbol{y} - \boldsymbol{y}'\|. \tag{5.11}$$

Define

$$\theta_x = \max\left\{\frac{1}{|\alpha_0|}, \frac{1}{|\alpha_1|}\right\} \tag{5.12}$$

$$\theta_y = \max\left\{\frac{\sum_{j=1}^N |\beta_{0,j}|}{|\alpha_0|}, \frac{\sum_{j=1}^N |\beta_{1,j}|}{|\alpha_1|}\right\}, \tag{5.13}$$

and

$$\Delta_{k-1} = \max\{\rho\Delta_k + q/2\,,\ \theta_x\Delta_k + \theta_y(\rho\Delta_k + q/2)\} \tag{5.14}$$

$$\tau_k = \pi_x\Delta_k + \pi_y(\rho\Delta_k + q/2)\,, \tag{5.15}$$

with $\Delta_K = q/2$. Assume that T_k satisfies

$$\big(p(A_1^{-1})^{-1} - p(A_0)\big)T_k \ge 2\tau_k\,, \tag{5.16}$$

and choose a threshold $\hat{T}_k$ such that

$$p(A_0)T_k + \tau_k \le \hat{T}_k \le p(A_1^{-1})^{-1}T_k - \tau_k\,. \tag{5.17}$$

Then we have

$$d_{\boldsymbol{n}}^k = [p\big(\hat{\boldsymbol{v}}^k(\boldsymbol{n})\big) > \hat{T}_k] \qquad \text{for } k = 1, \ldots, K\,, \tag{5.18}$$

and

$$\|\hat{x}^k - x^k\| \le \Delta_k \qquad \text{for } k = 0, \ldots, K\,. \tag{5.19}$$

Proof. We will prove this theorem by induction. First observe that (5.19) holds for $k = K$, i.e.,

$$\|\hat{x}^K - x^K\| \leq \Delta_K = \frac{q}{2}\,.$$

Now assume that (5.19) holds for some k with $1 \leq k \leq K$. We show that (5.18) holds for k and that we can also establish (5.19) for $k-1$. Thus, by induction, we conclude that, indeed, (5.18) holds for $k = 1, \dots, K$ and that (5.19) holds for $k = 0, \dots, K$.

Since $\hat{y}_j^k(\boldsymbol{n}) = Q^{-1}Q(y_j^k(\boldsymbol{n}))$, we get immediately that

$$\|\hat{\boldsymbol{y}}^k - \boldsymbol{y}^k\| \leq \frac{q}{2}, \quad k = 1, \dots, K\,.$$

We know that

$$\hat{y}_0^k(\boldsymbol{n}) = \hat{y}^k(\boldsymbol{n}) + P(\hat{x}^k)(\boldsymbol{n}) \qquad \text{and} \qquad y_0^k(\boldsymbol{n}) = y^k(\boldsymbol{n}) + P(x^k)(\boldsymbol{n})\,,$$

and from the estimate in (5.10) we get

$$\|\hat{\boldsymbol{y}}_0^k - \boldsymbol{y}_0^k\| \leq \rho\|\hat{x}^k - x^k\| + q/2 \leq \rho\Delta_k + q/2\,. \tag{5.20}$$

We now investigate the two possibilities for the decision map, namely $d_{\boldsymbol{n}}^k = 0$ and $d_{\boldsymbol{n}}^k = 1$. To simplify notation we suppress the argument $\boldsymbol{n}$.

(*i*) $d^k = 0$: this means that $p(\boldsymbol{v}_0^k) \leq T_k$ and we know from (5.6) that

$$\boldsymbol{v}^k = A_0\boldsymbol{v}_0^k$$

in this case. We derive an upper estimate for $p(\hat{\boldsymbol{v}}^k)$. Here we use that p satisfies the triangle inequality:

$$\begin{aligned} p(\hat{\boldsymbol{v}}^k) &\leq p(\boldsymbol{v}^k) + p(\hat{\boldsymbol{v}}^k - \boldsymbol{v}^k) = p(A_0\boldsymbol{v}_0^k) + p(\hat{\boldsymbol{v}}^k - \boldsymbol{v}^k) \\ &\leq p(A_0)p(\boldsymbol{v}_0^k) + p(\hat{\boldsymbol{v}}^k - \boldsymbol{v}^k) \\ &\leq p(A_0)T_k + \pi_x\|\hat{x}^k - x^k\| + \pi_y\|\hat{\boldsymbol{y}}_0^k - \boldsymbol{y}_0^k\|\,, \end{aligned}$$

where we have used estimate (5.11). Thus, we get

$$p(\hat{\boldsymbol{v}}^k) \leq p(A_0)T_k + \pi_x\Delta_k + \pi_y(\rho\Delta_k + q/2) \leq \hat{T}_k\,.$$

(*ii*) $d^k = 1$: this means that $p(\boldsymbol{v}_0^k) > T_k$ and we know from (5.6) that

$$\boldsymbol{v}^k = A_1\boldsymbol{v}_0^k$$

in this case. We derive a lower estimate for $p(\hat{\boldsymbol{v}}^k)$:

$$\begin{aligned} p(\hat{\boldsymbol{v}}^k) &\geq p(\boldsymbol{v}^k) - p(\hat{\boldsymbol{v}}^k - \boldsymbol{v}^k) = p(A_1\boldsymbol{v}_0^k) - p(\hat{\boldsymbol{v}}^k - \boldsymbol{v}^k) \\ &\geq p(A_1^{-1})^{-1}p(\boldsymbol{v}_0^k) - p(\hat{\boldsymbol{v}}^k - \boldsymbol{v}^k) \\ &> p(A_1^{-1})^{-1}T_k - \pi_x\|\hat{x}^k - x^k\| - \pi_y\|\hat{\boldsymbol{y}}_0^k - \boldsymbol{y}_0^k\|\,, \end{aligned}$$

and from (5.20) we arrive at the estimate

$$p(\hat{\boldsymbol{v}}^k) > p(A_1^{-1})^{-1}T_k - \pi_x\Delta_k - \pi_y(\rho\Delta_k + q/2) \geq \hat{T}_k\,.$$

Thus we conclude that

$$d^k = [p(\hat{\boldsymbol{v}}^k) > \hat{T}_k]\,.$$

From $\hat{x}_0^k(\boldsymbol{n}) = \frac{1}{\alpha_d}\big(\hat{x}^k(\boldsymbol{n}) - \sum\limits_{j=1}^{N} \beta_{d,j}\hat{y}_{0,j}^k(\boldsymbol{n})\big)$ and $x_0^k(\boldsymbol{n}) = \frac{1}{\alpha_d}\big(x^k(\boldsymbol{n}) - \sum\limits_{j=1}^{N} \beta_{d,j}y_{0,j}^k(\boldsymbol{n})\big)$, we can show that

$$\|\hat{x}_0^k - x_0^k\| \leq \theta_x\|\hat{x}^k - x^k\| + \theta_y\|\hat{\boldsymbol{y}}_0^k - \boldsymbol{y}_0^k\|\,.$$

Using (5.20) we arrive at

$$\|\hat{x}_0^k - x_0^k\| \leq \theta_x\Delta_k + \theta_y(\rho\Delta_k + q/2)\,. \tag{5.21}$$

Now we need one final step to compute $\hat{x}^{k-1}$, namely the merging of $\hat{x}_0^k$ and $\hat{y}_0^k$ by means of S^{-1}. It is obvious that

$$\|\hat{x}^{k-1} - x^{k-1}\| \leq \max\{\|\hat{x}_0^k - x_0^k\|, \|\hat{\boldsymbol{y}}_0^k - \boldsymbol{y}_0^k\|\}\,.$$

With (5.20) and (5.21) this yields

$$\begin{aligned} \|\hat{x}^{k-1} - x^{k-1}\| &\leq \max\{\rho\Delta_k + q/2, \theta_x\Delta_k + \theta_y(\rho\Delta_k + q/2)\} \\ &= \Delta_{k-1}\,. \end{aligned}$$

This proves the result. □

5.6 A case study

We consider the case where the splitting S in Fig. 5.3 corresponds with the quincunx polyphase decomposition in $\mathbb{Z}^2$ and where $p(\boldsymbol{v}) = |\boldsymbol{a}^T\boldsymbol{v}|$ with $\boldsymbol{a}^T\boldsymbol{u} \neq \boldsymbol{0}$. In order to satisfy the threshold criterion we need (see Proposition 4.1.2):

$$|\alpha_0| \leq |\alpha_1| \quad \text{and} \quad \boldsymbol{\beta}_d = \gamma_d\boldsymbol{a}\,,\ \gamma_d \in \mathbb{R}, \quad \text{for} \quad d = 0, 1\,. \tag{5.22}$$

Now, (5.16) yields

$$(|\alpha_1| - |\alpha_0|)T_k \geq 2\tau_k\,, \tag{5.23}$$

and from (5.12)-(5.13) we get that

$$\theta_x = |\alpha_0|^{-1} \quad \text{and} \quad \theta_y = \frac{\sum_{j=1}^N |a_j|}{|\sum_{j=1}^N a_j|} \cdot \max\Big\{|\frac{1-\alpha_0}{\alpha_0}|, |\frac{1-\alpha_1}{\alpha_1}|\Big\}\,.$$

Here we have used (5.22) and the fact that $\alpha_d + \gamma_d\sum_{j=1}^N a_j = 1$.

We consider $N = 4$ and choose samples y_j such that they correspond with the four horizontal and vertical neighbors of an 'even' location $\boldsymbol{n}$. The prediction of y_j is computed by averaging its four horizontal and vertical neighbors and we get that $\rho = 1$ in this case. We take $\boldsymbol{a} = (1,1,1,1)^T$ (i.e., the seminorm p models the Laplacian operator), $\alpha_1 = 1$ and $\alpha_0 = 2/3$. One can easily find that $\pi_x = \pi_y = 4$, $\theta_x = 3/2$ and $\theta_y = 1/2$. Thus, we get from (5.15) that $\tau_k = 8\Delta_k + 2q$, and from (5.14) that $\Delta_{k-1} = 2\Delta_k + q/4$, for $k = 1, \ldots, K$. Furthermore, we can show that

$$\Delta_k = \left(3 \cdot 2^{K-k} - 1\right)\frac{q}{4}\,.$$

Then, (5.23) can be rewritten as

$$T_k \geq 6\tau_k = 48\Delta_k + 12q = 36q \cdot 2^{K-k}\,. \tag{5.24}$$

Note that lower levels require higher thresholds, and that the larger is K, the larger the thresholds. If we want to recover all the decisions made at analysis, one should satisfy the above condition and choose $\hat{T}_k$ according to (5.17) (e.g., if $T_k = 6\tau_k$, then $\hat{T}_k = 5\tau_k$). Note, however, that for large quantization steps (large q), condition (5.24) is somewhat restrictive, especially for finer levels (lower k). For example, if $K = 4$, then $T_1 \geq 288q$ which, even for small q, will result in $d = 0$ for almost all decisions in the first level of decomposition. Thus, if the image is smooth enough, the adaptive wavelet scheme will behave exactly, or very similar, to the non-adaptive scheme with fixed $d = 0$.

First, we show some results where (5.24) is not satisfied (hence in the reconstruction we may have 'wrong' decisions). Nonetheless, the simulations results indicate that the adaptive scheme attains not only higher coding gain but also better visual quality in comparison with the non-adaptive scheme.

For the simulations we use the set of images shown at Fig. 5.1-Fig. 5.2. Unless otherwise stated, we use four levels of decompositions (i.e., $K = 4$) and a fixed threshold $T = 40$ for all levels. The images are decomposed with the adaptive wavelet scheme described above. The coarsest approximation x^4 and the details $y^1, \ldots, y^4$ are then uniformly quantized with some quantization step $q \geq 1$ (e.g., $\tilde{y}^k(\boldsymbol{n}) = \{y^k(\boldsymbol{n})/q\}$). For simplicity, we do not use any coding scheme. Thus, at synthesis the quantized transform coefficients are de-quantized (e.g., $\hat{y}^k(\boldsymbol{n}) = q \cdot \tilde{y}^k(\boldsymbol{n})$) and the inverse wavelet transform is applied. As a performance measure, we use the PSNR defined in (5.2) and the bit rate as given by the entropy defined in (5.9). We repeat this compression/decompression process with different quantization steps in order to compute the rate-distortion curves (bit rate versus PSNR). For comparison purposes, we perform the same experiments using the non-adaptive wavelet decomposition with fixed $d = 0$.

Fig. 5.5 shows the rate-distortion curves of the adaptive wavelet decomposition against the non-adaptive one, for the images in Fig. 5.1. One can see that the adaptive scheme has in general better PSNR performance.

Fig. 5.6-Fig. 5.7 show the reconstructed 'Lenna' images at rate 0.5 and 0.28 bpp, respectively, for both adaptive (left) and non-adaptive (right) schemes. Because of the adaptive filtering, the left images are less blurred than the right ones, especially for high-frequency textures. To appreciate the differences we show zooms of the hat, shoulder and feathers.

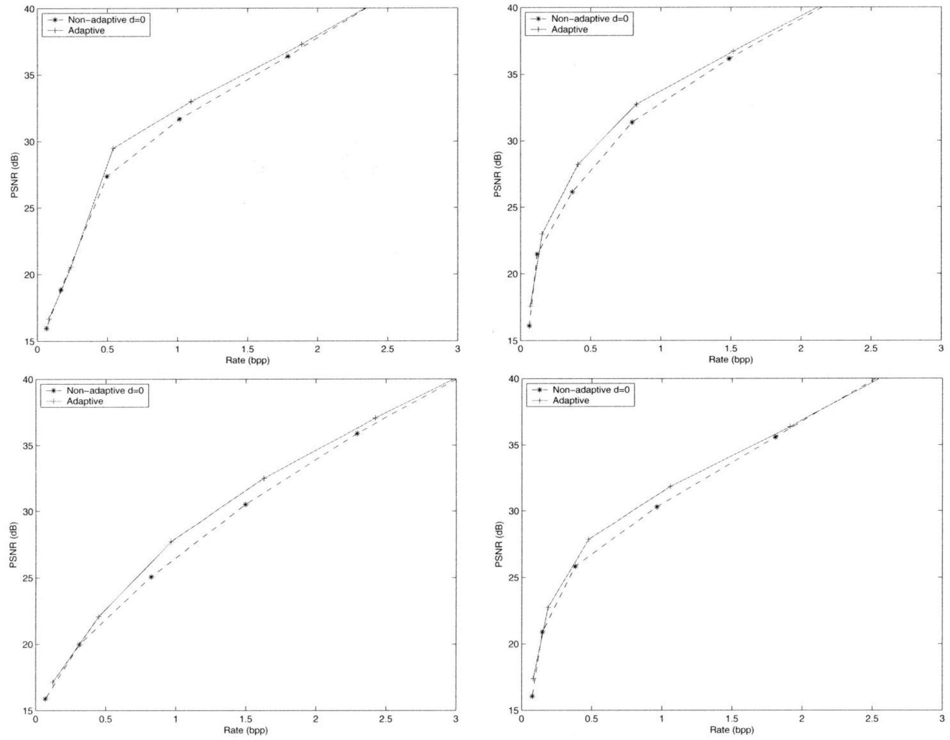


Figure 5.5: *Rate-distortion curves for test images in Fig. 5.1. From left to right and from top to bottom: 'House', 'Lenna', 'Barbara' and 'Goldhill'. The solid curve corresponds with the adaptive scheme, the dashed curve corresponds with the non-adaptive scheme with* $d = 0$.

In Fig. 5.8, we see the result of the adaptive scheme on the synthetic images of Fig. 5.2 compressed to 0.31 bpp. We note that in the non-adaptive case, the images suffer from blurring and ringing around the edges. In the adaptive case, ringing is reduced and edges are better preserved.

In Fig. 5.9, we show the different levels of reconstructed approximation 'Lenna' images with the adaptive scheme. Wavelet aliasing artifacts are quite noticeable in the early stages (lower levels) of the reconstruction. The blocking effect is partly due to the quincunx sampling scheme.

Finally, we give some examples where conditions in Proposition 5.5.1 (and hence (5.24)) are satisfied. This means that the decisions made at analysis can be perfectly recovered. Using the same update filters, we take $q = 2$ and $T_k = 72 \cdot 2^{(4-k)}$, $k = 1, \ldots, 4$. Hence, we have to use the synthesis thresholds $\hat{T}_k = 60 \cdot 2^{(4-k)}$, $k = 1, \ldots, 4$.

The top row of Fig. 5.10 shows the reconstructed images for both adaptive (left) and non-adaptive (right) wavelet decompositions applied to 'Barbara' image. The middle row shows the decision maps at level $k = 2$ and $k = 4$ used in the adaptive case. Note that the decision map at

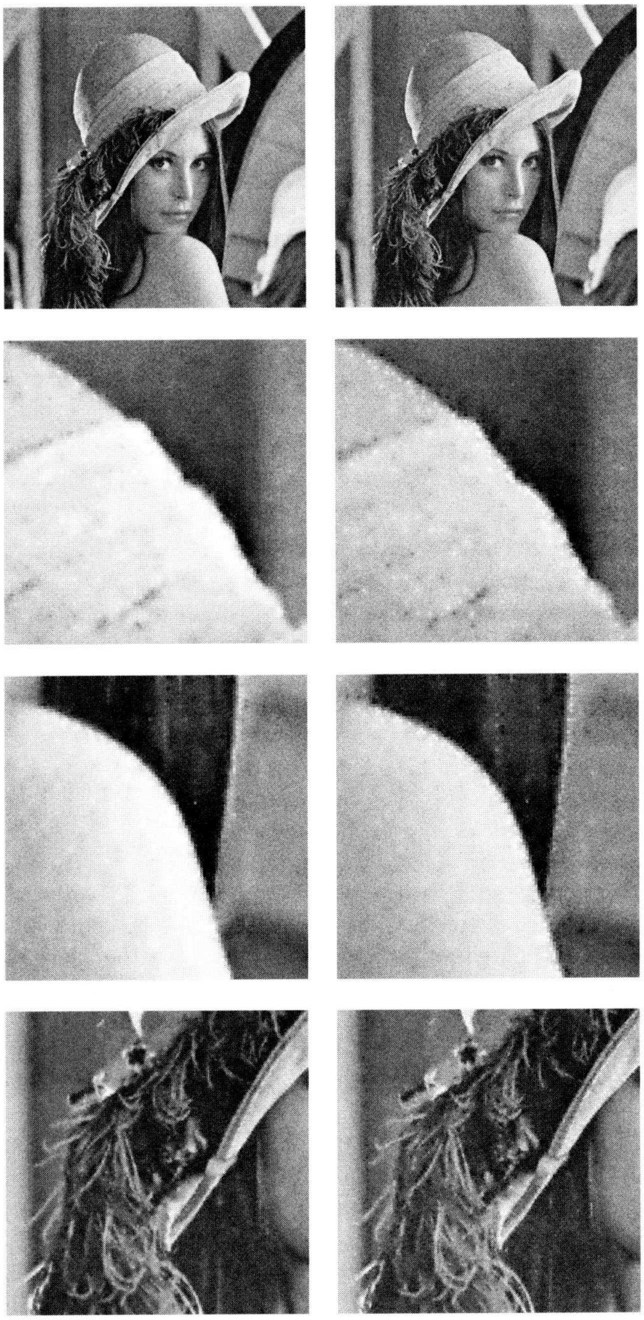


Figure 5.6: *Reconstructed images at bit rate* 0.5 *bpp for adaptive (left) and non-adaptive (right) schemes.*

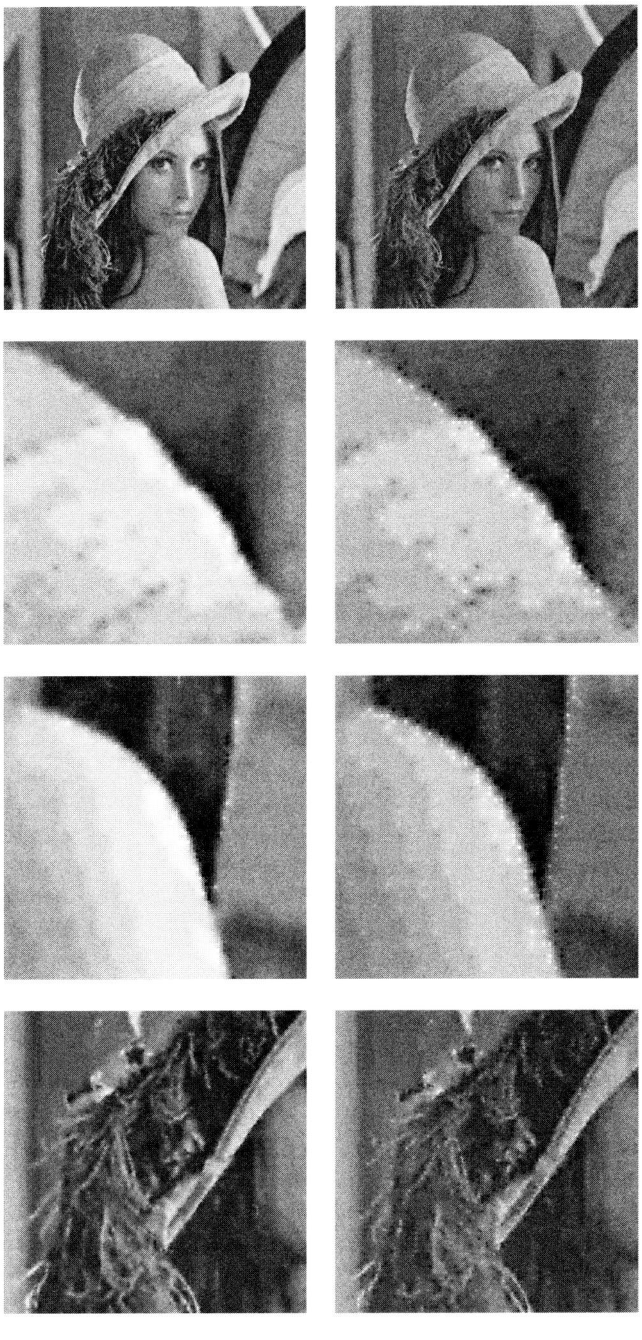


Figure 5.7: *Reconstructed images at bit rate* 0.28 *bpp for adaptive (left) and non-adaptive (right) schemes.*

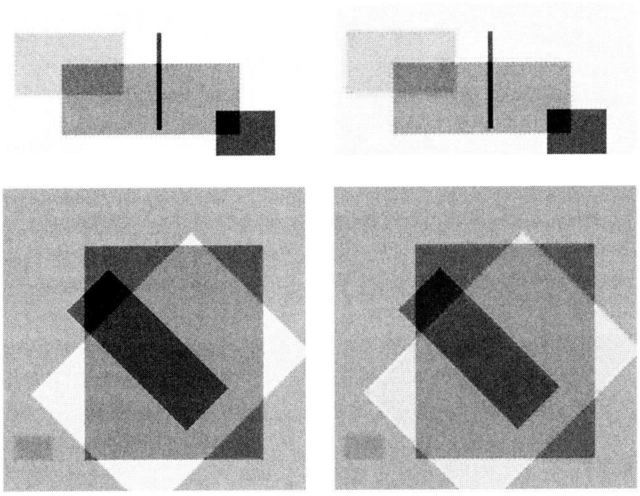


Figure 5.8: *Reconstructed images at* 0.31 *bpp for adaptive (left) and non-adaptive (right) schemes.*

Figure 5.9: *Progressive decoding. From left to right:* $\hat{x}^4, \hat{x}^2$ *and* $\hat{x}^0$.

level 2 (and the same applies to level 1 not shown here) has few values equal to 1. This means that at those levels the system behaves mostly as the non-adaptive scheme with $d = 0$. Although the reconstructed images with the adaptive decomposition are visually indistinguishable from the non-adaptive, we find that the residual error between the original and the reconstructed (see bottom row) has 5% less energy than in the non-adaptive case.

Fig. 5.11 shows the same experiment with 'Lenna' image. Note that here the decision map at level 2 has practically all values equal to 0, and that even at the highest level $k = 4$, the decision map has few values equal to 1. The energy of the residual error for the adaptive decomposition is 1.3% less than for the non-adaptive one.

From Fig. 5.10-Fig. 5.11, one can see that the differences between the adaptive and non-adaptive scheme are negligible due to the high thresholds that have been used ($T_1 = 576$, $T_2 = 288$, $T_3 = 144$ and $T_4 = 72$). One can achieve better performances by lowering the thresholds (see e.g. Fig. 5.5-Fig. 5.9), though this might[3] imply that the decision map at synthesis is not exactly the same as the one at analysis.

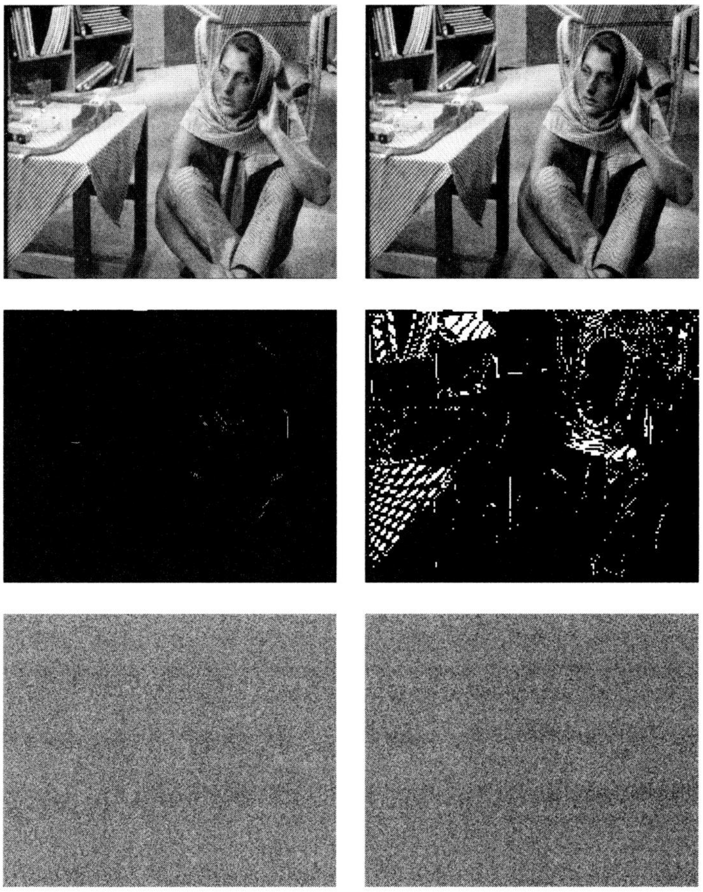


Figure 5.10: *Top: reconstructed images for the adaptive (left) and non-adaptive (right) schemes. Middle: decision map at levels 2 (left) and 4 (right). Bottom: error images for the adaptive (left) and non-adaptive (right). Here, $q = 2$ and conditions in Proposition 5.5.1 are satisfied.*

Finally, we perform the same experiment but with $K = 2$ and $q = 1$ for the synthetic image 'Synth2'. The results are shown in Fig. 5.12. In this case, the reconstruction error resulting

---

[3] Note that conditions in Proposition 5.5.1 were obtained for the worst of the cases. In practice, one can reconstruct the decisions under milder conditions.

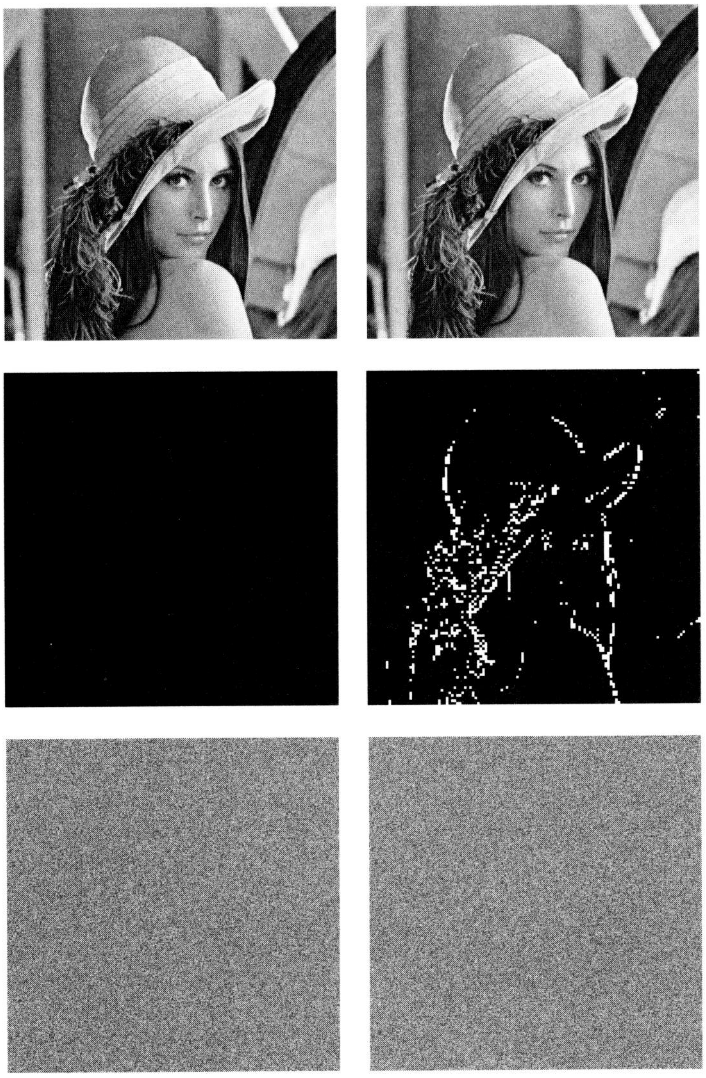


Figure 5.11: *Top: reconstructed images for the adaptive (left) and non-adaptive (right) schemes. Middle: decision map at levels 2 (left) and 4 (right). Bottom: error images for the adaptive (left) and non-adaptive (right) schemes. Here,* $q = 2$ *and conditions in Proposition 5.5.1 are satisfied.*

from the adaptive decomposition (bottom left of Fig. 5.12) has much less energy (64% less) than the error obtained in the non-adaptive case (bottom right of Fig. 5.12). This example shows again (see Table 5.3) that our adaptive wavelet decompositions are especially suited to discriminate between homogeneous regions and sharp transitions (edges).

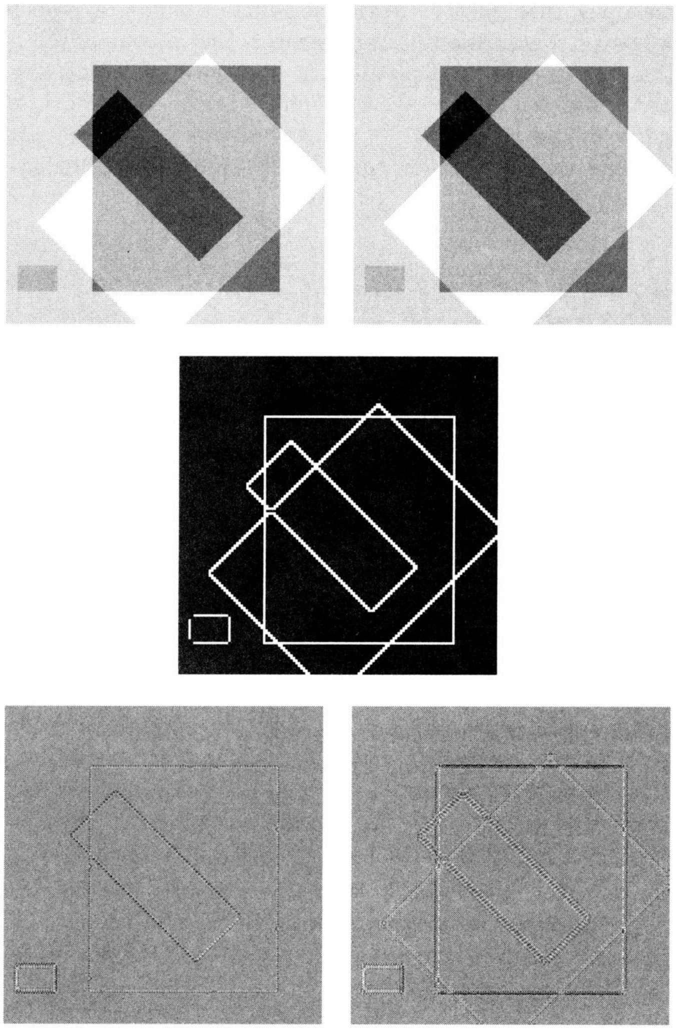


Figure 5.12: *Top: reconstructed images for the adaptive (left) and non-adaptive (right) schemes. Middle: decision map at level 2. Bottom: error images for the adaptive (left) and non-adaptive (right) schemes. Here, $q = 1$, $K = 2$ and conditions in Proposition 5.5.1 are satisfied.*

5.7 Discussion

In this chapter we have discussed the use of the adaptive wavelet transforms proposed in Chapter 3 for image compression. The results that have been obtained are quite encouraging.

It has been shown (Section 5.4) that the adaptive wavelet schemes yield decompositions that have lower entropies than schemes with fixed update filters, a property that is highly relevant in the context of compression. Note, however, that a comparison based only on the entropies is questionable. First, (in a progressive transmission context) the visual 'quality' of the approximation images must be taken into account. In Chapter 4, we have seen that the subjective quality of the approximation images obtained with the adaptive schemes is significantly better than that of the non-adaptive schemes. Here, quality refers to lack of blurring and edge sharpness. Secondly, for real compression schemes, quantization *and* coding must be applied to the resulting decompositions in order to exploit the residual redundancies.

In Section 5.5, we have analyzed the quantization effects on the adaptive scheme. We have been able to derive conditions that guarantee perfect reconstruction of the decision map after quantization. A particular case has been worked out in Section 5.6. We have obtained rate-distortion curves that indicate an improvement of the adaptive scheme over the non-adaptive for low bit rates. Overall, the simulation results indicate that the adaptive schemes can attain higher coding gain as well as better visual quality than their non-adaptive counterparts. The reason for the improvements is that edges in the adaptive decomposition are represented in a more compact form, and as a result there is less degradation of the reconstructed image when discarding small, non-zero coefficients.

Note that we have not used any coding step. Thus, bit rates (used in the rate-distortion curves) are calculated from entropy estimates, not the actual encoding size. We expect that embedding the adaptive wavelet decomposition into existing coders such as EZW will result in further improvements.

Observe also that we gave up the use of different quantization steps for each band, to achieve lower complexity. The use of different quantization steps as well as decision criteria for different bands is currently under investigation [71].

Chapter 6

Multiresolution image fusion

Extraordinary advances in sensor technology, microelectronics and communications have brought a need for processing techniques that can effectively combine information from different sources into a single composite for interpretation. In image-based application fields, image fusion has emerged as a promising research area.

Image fusion provides the means to integrate multiple images into a composite image that is more suitable for the purposes of human visual perception and computer-processing tasks such as segmentation, feature extraction and target recognition. For example, the fusion of visual and infrared images in an airborne sensor can aid pilots navigate in poor weather conditions, and the fusion of computer tomography and magnetic resonance images may facilitate medical diagnosis.

Among the various frameworks in which image fusion has been formulated, the multiresolution approach is one of the most intensively studied and used in practice. In this chapter, we reframe the multiresolution-based fusion methodology into a common formalism which encompasses most of the existing multiresolution fusion schemes and provides freedom to create new ones. After a brief introduction to image fusion in Section 6.1, a general framework for pixel-based multiresolution fusion is presented in Section 6.2. Within this framework, some of the existing schemes are described in Section 6.3. Fusion results examples of existing as well as new fusion schemes are shown in Section 6.4. Finally, in Section 6.5, we study how the adaptive lifting scheme proposed in Chapter 3 can be used in multiresolution image fusion.

6.1 Image fusion

In this section, we introduce the concept of image fusion. We discuss the rationale behind fusion, and the various issues that need to be addressed when designing a fusion scheme. We outline some applications and review some of the most important fusion techniques used in practice.

6.1.1 Concept of image fusion

Image fusion[1] can be broadly defined as the process of combining multiple input images into a smaller collection of images, usually a single one, which contains the 'relevant' information from the inputs, in order to enable a good understanding of the scene, not only in terms of position and geometry, but more importantly, in terms of semantic interpretation. In this context, the word 'relevant' should be considered in the sense of 'relevant with respect to the task the output images will be subject to', in most cases high-level tasks such as interpretation or classification. In the sequel, we will refer to this 'relevant' information as *salient* information. The images to be combined will be referred to as *input* or *source* images, and the fusion result image (or images) as *composite* image.

The actual fusion process can take place at different levels of information representation. A common categorization is to distinguish between pixel, feature and decision level [96, 115], although there may be crossings between them. Image fusion at pixel level amounts to integration of low-level information, in most cases physical measurements such as intensity [123, 165]. It generates a composite image in which each pixel is determined from a set of corresponding pixels in the various sources. Fusion at feature level requires first the extraction (e.g., by segmentation procedures) of the features[2] contained in the various input sources [12, 42]. Those features can be identified by characteristics such as size, shape, contrast and texture. The fusion is thus based on those extracted features and enables the detection of useful features with higher confidence. Fusion at decision level allows the combination of information at the highest level of abstraction [41, 78]. The input images are usually processed individually for information extraction and classification. This results in a number of symbolic representations which are then fused according to decision rules which reinforce common interpretation and resolve differences. The choice of the appropriate level depends on many different factors such as the characteristics of the physical sources, the specific application and the tools that are available. At the same time, the choice of the fusion level determines the pre-processing that is required. For instance, fusion at pixel level (*pixel fusion*) requires co-registered images at subpixel accuracy because pixel fusion methods are very sensitive to misregistration.

Today, most image fusion applications employ pixel fusion methods. The advantage of pixel fusion is that the images used contain the original information. Furthermore, the algorithms are rather easy to implement and time efficient. As we observed before, an important preprocessing step in pixel fusion methods is image registration, which ensures that the data at each source is referring to the same physical structures. In the remainder, it will be assumed that all source images have been registered. Comprehensive reviews on image registration can be found in [13, 91, 151, 161].

---

[1] Terminologies such as fusion, integration and merging, are often used interchangeably in the literature.

[2] A feature is any distinguishing property or attribute of an image. Examples of features used in image fusion are edges, lines, patterns, color, etc.

6.1.2 Objectives, requirements and challenges of image fusion

The aim of image fusion is to integrate complementary and redundant information from multiple images to create a composite that contains a 'better' description of the scene than any of the individual source images. Utilization of the composite image is expected to increase the performance of the subsequent processing tasks. By integrating information, image fusion can reduce dimensionality. This results in a more efficient storage and faster interpretation of the output. By using redundant information, image fusion may improve accuracy as well as reliability, and by using complementary information, image fusion may improve interpretation capabilities with respect to subsequent tasks. This leads to more accurate data, increased utility and robust performance. Considering the objectives of image fusion and its potential advantages, some generic requirements can be imposed on the the fusion algorithm [123]:

- it should not discard any salient information contained in any of the input images;
- it should not introduce any artifacts or inconsistencies which can distract or mislead a human observer or any subsequent image processing steps;
- it must be reliable, robust and, as much as possible, tolerant of imperfections such as noise or misregistrations.

Clearly, a choice as to which information is salient has to be made. Here again, knowledge about input data and application plays a crucial role. However, a fusion approach which is independent of the modalities of the inputs and produces a composite image which appears 'natural' to a human interpreter, is highly desirable.

The requirements listed above are often very difficult to achieve and even more difficult to assess. The problem of evaluating image fusion methods lies in the variety of different application requirements and the lack of a clearly defined ground-truth. The topic of performance evaluation will be discussed in more detail in Chapter 8.

To illustrate some of the challenges we have to face when developing a fusion algorithm, consider the source images in Fig. 6.1 depicting the same scene. While in the visual image of Fig. 6.1(a) it is hard to distinguish the person in camouflage from the background, this person is clearly observable in the infrared (IR) image of Fig. 6.1(b). In contrast, the easily discernible background in the visual image, such as the fence, is nearly imperceptible in the IR image. Now the question is how to combine both images in a unique composite which represents the overall scene better than any of the two individual images. We sum up explicitly some of the difficulties that we encounter:

- Complementary information: some image features appear in one source but not in the other, e.g., the person in Fig. 6.1(b) or the fence in Fig. 6.1(a).
- Common but contrast reversal information: there are various objects and regions that occur in both images but with opposite contrast, e.g., part of the roof of the house or the bushes at the left lower corner. Thus, the direct approach of averaging the source images is not satisfactory.

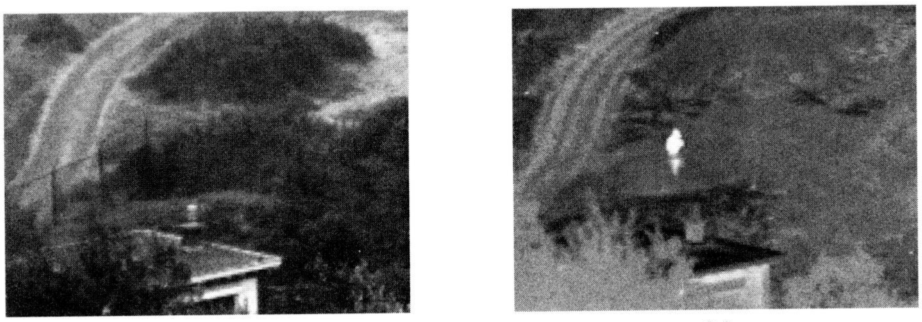

(a) (b)

Figure 6.1: *Example of source images to be fused: (a) visual image; (b) infrared image. Images courtesy of Alexander Toet, from TNO Human Factors Institute, The Netherlands.*

- Disparity between sensors: input images come from different types of sensors which may have different dynamic range and different resolution. Moreover, they may not be equally reliable. If possible, such disparities have to be taken into account when comparing the content of the information in the images.

This is, by no means, an exhaustive list of problems that could arise. Furthermore, we should also be aware of the inherent difficulties present in any image acquisition and analysis task: presence of noise, sensor calibration or hardware limitations, to name a few.

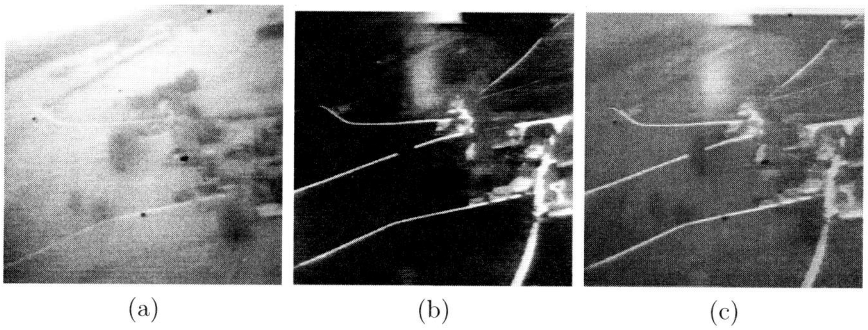
6.1.3 Application fields

Image fusion is widely recognized as a valuable tool for improving overall system performance in image-based application areas such as defense surveillance, remote sensing, medical imaging and computer vision. We list some application fields and give some references to the relevant literature.

Military

Historically, military appears to be the first application area for image fusion. It covers subareas such as detection, identification and tracking of targets [6,10,149], mine detection [106, 108], tactical situation assessment [124,157], and person authentication [109].

Fig. 6.2 illustrates how information from visible and IR wavelength images can improve situational awareness in a typical pilotage scene. Note that the IR image in Fig. 6.2(b) contains much of the road network details while the visual image in Fig. 6.2(a) provides horizon information and additional building and vegetation details. Note also that the light spots appear only in the visual image, where they are perceived as small dark blobs. Of different nature is the glare effect on the IR image. This is due to common scanner interference and is usually perceived as a ripple effect. The composite image, shown in Fig. 6.2(c), contains the most salient information from each sensor.



(a) (b) (c)

Figure 6.2: *Fusion of visual and IR images: (a) visual image; (b) IR image; (c) composite image obtained by a multiresolution fusion strategy (Section 6.2). Here, a discrete wavelet transform (2 levels, Daubechies (2,2)) and a maximum selection rule were employed.*

Geoscience

This field concerns the earth study with satellite and aerial images (remote sensing) [115, 119]. A major problem is interpretation and classification of images. The fusion of images from multiple sensors allows the detection of roads, airports, mountainous areas, etc. [36, 89, 159].

In remote sensing applications, there is often a difference in spatial or wavelength resolution between the images produced by different sensors. A typical example is the merging of a high-resolution SPOT Panchromatic image with Landsat Thematic Mapper multispectral images. The Landsat spectral bands enable classification of objects and areas in the scene, while the high spatial resolution SPOT band provides a more accurate localization of the observed objects. A major challenge is to preserve the higher spatial resolution of the SPOT band without destroying the spectral information content provided by the Landsat bands [35, 120].

Fig. 6.3 exemplifies the fusion of two bands of a multispectral scanner. Band 1 penetrates water and is useful for mapping along coastal areas, for soil-vegetation differentiation and for distinguishing forest types. In Fig. 6.3(a), buildings, roads and different agricultural zones are clearly discernible. Band 2 is more convenient for highlighting green vegetation and for detecting water-land interfaces. In Fig. 6.3(b), the bay is sharply delineated. The composite image in Fig. 6.3(c) contributes to a better understanding of the objects observed and allows a more accurate identification.

Medical imaging

Fusion of multimodal images can be very useful for clinical applications such as diagnosis, modeling of the human body or treatment planning [74, 103, 104, 171].

The next example illustrates the usage of fusion in radiotherapy and skull surgery. Here, the information provided by magnetic resonance imaging (MRI) and X-ray computed tomography (CT) is complementary. Normal and pathological soft tissues are better visualized by MRI (Fig. 6.4(a)), while the structure of tissue bone is better visualized by CT (Fig. 6.4(b)). The composite image, depicted in Fig. 6.4(c), not only provides salient information from both images simultaneously, but also reveals the relative position of soft tissue with respect to the bone

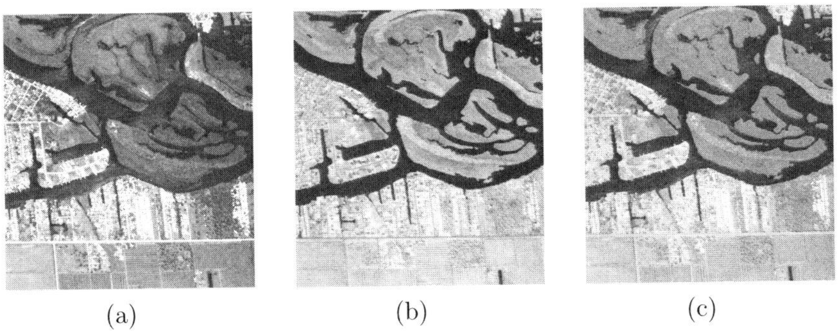

(a) (b) (c)

Figure 6.3: *Fusion of multispectral images: (a) image from band 1; (b) image from band 2; (c) composite image obtained by a multiresolution fusion strategy (Section 6.2). In this case, a shift-invariant discrete wavelet transform (2 levels, Daubechies (2,2)) and a maximum selection rule were employed.*

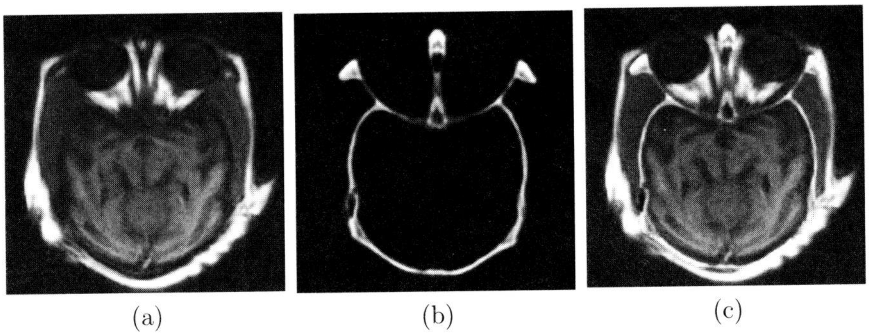

(a) (b) (c)

Figure 6.4: *Fusion of MRI and CT images: (a) MRI image; (b) CT image; (c) composite image obtained by a multiresolution fusion strategy (Section 6.2). Here, a morphological pyramid (2 levels, opening-closing and closing-opening filters) and a maximum selection rule were employed.*

structure.

Robotics and industrial engineering

Here, fusion is commonly used to identify the environment in which the robot or intelligent system evolves [2, 25]. It is also employed for navigation in order to avoid collisions and keep track of the trajectory [79, 112]. Image fusion is also employed in industry, for example, for the monitoring of factories or production lines [95], or for quality and defect inspection of products [121].

Fig. 6.5 shows how fusion can be used to extend the effective depth of field[3] of a vision system. Due to the limited depth of field of optical lenses, it is often not possible to get an

---

[3] The depth of field is the range of distance from a camera that is acceptably sharp in the image obtained by that camera.

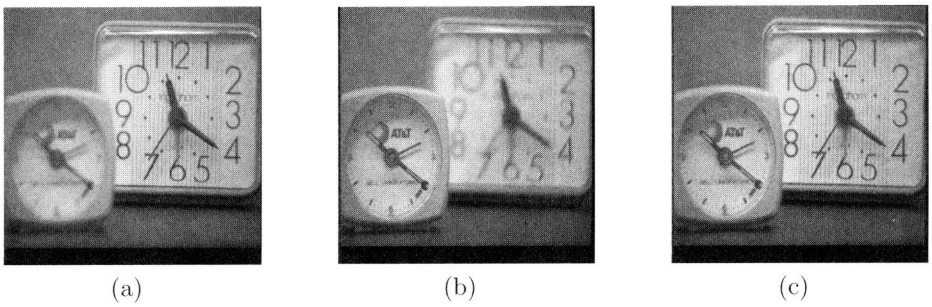

(a) (b) (c)

Figure 6.5: *Fusion of out-of-focus images: (a) image with focus on the right; (b) image with focus on the left; (c) composite image obtained by a multiresolution fusion strategy (Section 6.2). In this case, a Laplacian pyramid (3 levels) and a maximum selection rule were employed.*

image with all objects in focus. One way to overcome this problem is to take several recordings with different focus points and combine them into a single composite which contains the focused regions of all input images. This could be useful, for example, in digital camera design or in industrial inspection applications where the need to visualize objects at very short distances complicates the preservation of the depth of field.

6.1.4 Fusion techniques

There are various techniques for image fusion, even at the pixel level [115]. The selection of the appropriate one depends strongly on the type of application. Here, we outline some of the most commonly used techniques in pixel fusion. We have grouped them into four major categories; however, this is a rather loose classification since these categories do overlap in various ways.

Weighted combination

A simple approach for fusion consists of synthesizing the composite image by averaging corresponding pixels of the image sources. An 'optimal' weighting can be determined, for example, by a principal component analysis of the correlation or covariance matrix of the sources [122]. The weightings for each input are obtained from the eigenvector corresponding to the largest eigenvalue. Variations of this method and other arithmetic signal combinations are numerous [8, 87].

Color space fusion

Image fusion by color transformations takes advantage of the possibility of representing data in different color channels. The simplest technique is to map the data from a sensor to a particular color channel. Many different band combinations and color spaces can be applied [158, 167]. The challenge is to generate an intuitive, meaningful, color, composite image. Moreover, pseudocolor mappings can help to identify sensor-specific details in a composite image. That is, the use of color can be used to identify which sensor generated the features appearing in the composite image [158]. The benefits of false-color imagery relative to monochromatic and non-fused imagery in tasks such as detection and localization of targets have been studied in [156].

Optimization approach

This approach is based on an a priori model of the real scene and the fusion task is expressed as an optimization problem. In Bayesian optimization, the goal is to find the composite image which maximizes the a posteriori probability. Some examples of probabilistic fusion schemes can be found in [78, 135]. In the Markov random field approach, the input images are modeled as Markov random fields to define a cost function which describes the fusion goal [7]. A global optimization strategy such as simulated annealing can be employed to minimize this cost function.

Biology based approaches

One of the most famous examples of fusion in a living organism is the visual system of rattlesnakes [114]. These vipers possess organs which are sensitive to thermal radiation. The IR signals provided by these organs are combined by bimodal neurons with the visual information obtained from the eyes. Inspired by this real-world example, several researchers have used neural networks to model multisensor image fusion [56, 167].

Another biologically inspired fusion method is the approach based on multiresolution (MR) decompositions [3, 21, 27, 92, 154, 175]. It is motivated by the fact that the human visual system is primarily sensitive to local contrast changes, i.e. edges, and MR decompositions provide a convenient spatial-scale localization of such local changes. The basic strategy of a generic MR fusion scheme is to use specific fusion rules to construct a composite MR representation from the MR representations of the different input sources. The composite image is then obtained by performing the inverse decomposition process.

Henceforth, we confine our discussion to MR image fusion approaches. In particular, we focus on MR fusion schemes (working at pixel and feature level) where the output is a single composite image which is constructed primarily for display on a computer monitor.

6.2 A general pixel-based MR fusion scheme

The basic idea underlying the MR image fusion approach is to perform an MR transform on each source image and, following some specific fusion rules, construct a composite MR representation from these inputs. The composite image is obtained by applying the inverse transform on this composite MR representation. This process is illustrated in Fig. 6.6 for the case of two input images. Here, Ψ is the MR transform and Ψ^{-1} its inverse.

At the MR decomposition (analysis) stage, the data is transformed into a convenient representation which, besides scale or resolution, may also involve orientation or wavelength or some other physical parameters. At the combination stage, the actual fusion of the (transformed) data takes place. This involves identifying the salient information and transferring it into the composite image. This process, i.e., the combination of the data, is governed by a number of rules called the *fusion rules*. The result is a composite MR representation from which the composite image is obtained by application of the inverse MR transform (synthesis).

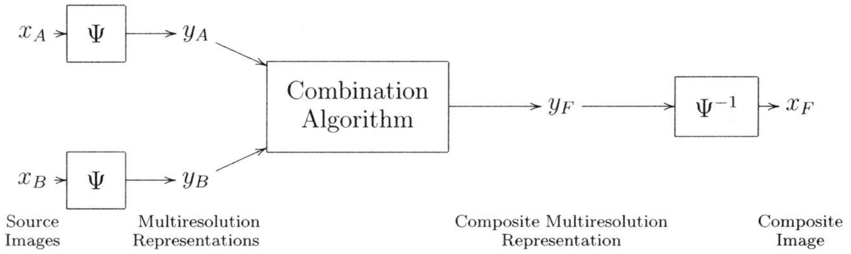


Figure 6.6: *MR image fusion scheme. Left: MR transform Ψ of the sources; middle: combination in the transform domain; right: inverse MR transform Ψ^{-1} of the composite representation.*

In the literature one finds several variants of the MR fusion scheme. In what follows, we present a general framework which encompasses most of them. Section 6.2.2 describes the various modules of the framework. Some of the existing algorithms that fit within our framework are reviewed in Section 6.3. Examples of such schemes as well as other implementation alternatives are given in Section 6.4.

It is to be noted that our fusion framework has been partially inspired by the MR fusion methodology proposed by Zhang and Blum in [175].

6.2.1 Notation

We fix some notation for MR image decompositions. As we have seen in Chapter 2, an input image x^0 can be represented by a sequence of detail images at different levels of resolution along with the coarsest approximation image. Henceforth, the MR decomposition of an image x^0 is denoted by y and it is assumed to be of the form:

$$y = \{y^1, y^2, \dots, y^K, x^K\}\,. \tag{6.1}$$

Here x^K represents the approximation image at the highest level (lowest resolution) of the MR structure, while images y^k, $k = 1, \dots, K$, represent the detail images at level k. The detail at level k will, in general, comprise various frequency or orientation bands, depending on the type of MR transform that has been used. We assume henceforth that y^k is composed of P detail images, i.e., $y^k = \{y^k(\cdot|1), \dots, y^k(\cdot|P)\}$.

Let I_x^k and $I_y^k(p)$ denote the domain of x^k and $y^k(\cdot|p)$ respectively. As in previous chapters, we use the vector coordinate $\boldsymbol{n} = (m, n)^T$ to index the location of the coefficient. Then, $x^k(\boldsymbol{n})$, where $\boldsymbol{n} \in I_x^k$, represents the approximation coefficient at location $\boldsymbol{n}$ within level k. Similarly, $y^k(\boldsymbol{n}|p)$, where $\boldsymbol{n} \in I_y^k(p)$, represents the detail coefficient at location $\boldsymbol{n}$ within level k and band p. Note that I_x^k is not necessarily equal to $I_y^k(p)$. In the pyramid case, for example, y^k represents a detail image of the same size as x^{k-1}, while in the standard wavelet transform $y^k(\cdot|p)$ is a detail image of the same dimensions as x^k. Note also that in most cases $I_y^k(p)$ does not depend on p.

For convenience, we will sometimes denote the approximation image x^k by $y^k(\cdot|0)$. In this way, we can use the general expression of $y^k(\cdot|p)$ to refer both to the detail images (for

$p = 1, \ldots, P$) and the approximation image (for $p = 0$). If no confusion is possible, we will use the shorthand notation $(\cdot)$ to denote $(\boldsymbol{n}|p)$; e.g., we will write $y^k(\cdot)$ rather than $y^k(\boldsymbol{n}|p)$.

6.2.2 The general framework

In Fig. 6.7 we show a more detailed version of the fusion scheme of Fig. 6.6, in which the combination algorithm has been specified. In our framework, the combination algorithm consists of four modules: the *activity* and *match* measures extract information from the MR decompositions of the input images, which is then used by the *decision*[4] and *combination* maps to compute the MR decomposition y_F of the composite image. Below, we give a short description of each of the building blocks. Note, however, that some of them, such as the 'match block', are optional.

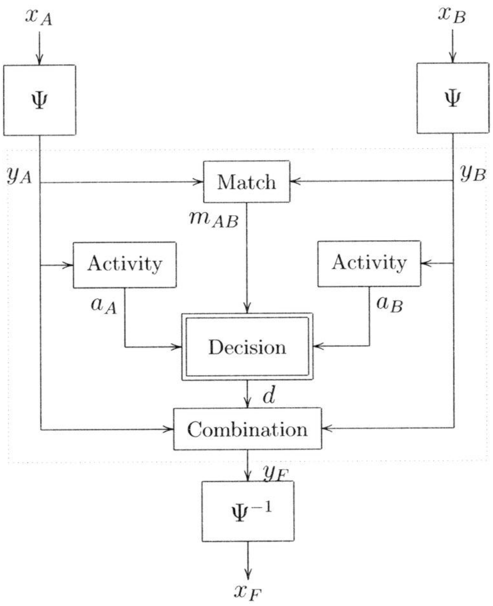


Figure 6.7: *Generic pixel-based MR fusion scheme with two input sources x_A and x_B, and one output composite image x_F.*

MR analysis (Ψ)

The analysis block computes an MR decomposition of the input sources x_S, $S \in \mathcal{S}$, where $\mathcal{S}$ is the index set of source images. For every input x_S we obtain its MR representation $y_S = \Psi(x_S)$, with y_S having the form defined in (6.1). That is,

$$y_S = \{y_S^1, y_S^2, \ldots, y_S^K, y_S^K(\cdot|0)\},$$

---

[4] This decision map has nothing to do with the decision map used in previous chapters.

where $y_S^K(\cdot|0)$ corresponds to the approximation image at the coarsest level K and $y_S^k = \{y_S^k(\cdot|p)\}$, $p = 1, \dots, P$, to the detail images at level k.

Activity measure

The degree of saliency of each coefficient in y_S (i.e., its importance for a task at hand) will be expressed by the so-called *activity*. The activity measure block associates to every band image $y_S^k(\cdot|p)$ an activity $a_S^k(\cdot|p)$, which reflects the local activity of the image.

Match measure

The match measure is supposed to quantify the degree of 'similarity' between the sources. More precisely, the match value $m_{AB}^k(\cdot)$ reflects the resemblance between the inputs $y_A^k(\cdot)$ and $y_B^k(\cdot)$.

Decision map

The decision map is the core of the combination algorithm. Its output governs the actual combination of the coefficients of the MR decompositions of the various sources. For each level k, orientation band p, and location $\boldsymbol{n}$, the decision map assigns a value $\delta = d^k(\boldsymbol{n}|p)$ which is then used for the computation of the composite $y_F^k(\boldsymbol{n}|p)$.

Combination map

The combination map describes the actual combination of the transform coefficients of the sources. For each level k, orientation band p, and location $\boldsymbol{n}$, the combination map yields the composite coefficient $y_F^k(\boldsymbol{n}|p)$.

MR synthesis (Ψ^{-1})

Finally, the composite image is obtained by applying the inverse transformation on the composite MR decomposition y_F, that is, $x_F = \Psi^{-1}(y_F)$, where Ψ^{-1} is the inverse MR transform.

From the previous description, one can see that the parameters and functions comprised by the different blocks can be chosen in several ways. In the following, we discuss them in more detail.

6.2.3 MR analysis and synthesis

As we have seen in Chapter 2, the MR representation y_S comprises information at different scales. High levels contain coarse scale information while low levels contain finer details. Such a representation is suitable for image fusion, not only because it enables one to consider and fuse image features separately at different scales, but also because it produces large coefficients near edges, thus revealing salient information [102].

Basically, the issues to be addressed are the specific type of MR decomposition (pyramid, wavelet, linear, morphological, etc.) and the number of decomposition levels.

A large part of research on MR image fusion has focused on choosing an appropriate MR representation which facilitates the selection and combination of salient features. After an exhaustive study of the existing literature, we have reached the following conclusions:

- In general, sampling causes a deterioration in the quality of the composite image by introducing heavier blocking effects than would have been obtained by using decompositions without sampling.
- Often, shift and rotation-invariance properties are required. For many applications, the fusion result should not depend on the location or orientation of the objects in the input sources. Shift and rotation dependency are especially undesirable considering misregistration problems and for image sequence fusion.
- In linear approaches, the specific filters used to implement the MR transform have little influence on the fusion result. In general, shorter filters lead to slightly sharper composite images.
- MR decompositions constructed with morphological techniques are more suited for the analysis of the shape and size of specific features in the images.

Another parameter which influences performance is the number of decomposition levels (analysis depth). To perform a consistent fusion of objects at arbitrary scales, decomposition over a large number of scales may appear necessary. However, using more levels does not necessarily yield better results; it may produce low-resolution bands where neighboring features overlap. This gives rise to discontinuities in the composite representation and thus introduces distortions such as blocking effects or 'ringing' artifacts into the composite image. The required analysis depth is primarily related to the spatial extent of the relevant objects in the source images. In general, it is not possible to compute the optimal analysis depth, but as a rule of thumb, the larger the objects of interest are, the higher the number of decomposition levels should be.

6.2.4 Activity measure

The meaning of 'saliency' (and thus the computation of the activity) depends on the nature of the source images as well as on the particular fusion application. For example, when combining images having different foci, a desirable activity level measure would provide a quantitative value that increases when features are more in focus. In this case, a suitable measure is one that puts emphasis on contrast differences. Since contrast information is partially captured in the decomposition by the magnitude of high-frequency components (details), a good choice is the absolute value of the detail coefficients or some other function that operates on their amplitude. Generally, based on the fact that the human visual system is primarily sensitive to local contrast changes (i.e., edges), most fusion algorithms compute the activity as some sort of energy calculation, e.g.,

$$a_S^k(\boldsymbol{n}|p) = \sum_{\Delta\boldsymbol{n}\in\mathcal{W}^k(p)} w^k(\Delta\boldsymbol{n}|p)\left|y_S^k(\boldsymbol{n}+\Delta\boldsymbol{n}|p)\right|^\gamma , \qquad \gamma \in \mathbb{R}_+ , \tag{6.2}$$

where $\mathcal{W}^k(p)$ is a finite window at level k and orientation p, and $w^k(\cdot|p)$ are the window's weights. In the simplest case, the activity is just the absolute value of the coefficient, that is,

$$a_S^k(\cdot) = \left|y_S^k(\cdot)\right| .$$

Alternatively, one can compute the activity as the contrast of the component with its neighbors, e.g.,

$$a_S^k(\boldsymbol{n}|p) = \frac{\left|y_S^k(\boldsymbol{n}|p)\right|}{\sum\limits_{\Delta\boldsymbol{n}\in\mathcal{W}^k(p)} w^k(\Delta\boldsymbol{n}|p)\left|y_S^k(\boldsymbol{n}+\Delta\boldsymbol{n}|p)\right|},$$

or using some other linear or nonlinear criteria. For instance, to reduce the influence of impulsive noise, one may consider

$$a_S^k(\boldsymbol{n}|p) = \mathrm{median}_{\Delta\boldsymbol{n}\in\mathcal{W}^k(p)}\left|y_S^k(\boldsymbol{n}+\Delta\boldsymbol{n}|p)\right|.$$

In practice, the window $\mathcal{W}^k(p)$ over which the function operates is small, typically including only the sample itself (*sample-based* operation), or a 3×3 or 5×5 window centered at the sample (*area-based* operation). However, other size and shape windows have also been used. Increasing the size of the neighborhood in the simple sample-based case adds robustness to the fusion system as it provides a smooth activity function. However, larger windows cause problems at lower resolution levels when their size exceeds the size of the most salient features.

6.2.5 Match measure

The match[5] or similarity between the transform coefficients of the source images is usually expressed in terms of a local correlation measure. Alternatively, the relative amplitude of the coefficients or some other criteria can be used. In the following expression, the match value between $y_A^k(\cdot)$ and $y_B^k(\cdot)$ is defined as a normalized correlation averaged over a neighborhood of the samples:

$$m_{AB}^k(\boldsymbol{n}|p) = \frac{2\sum\limits_{\Delta\boldsymbol{n}\in\mathcal{W}^k(p)} w^k(\Delta\boldsymbol{n}|p)y_A^k(\boldsymbol{n}+\Delta\boldsymbol{n}|p)y_B^k(\boldsymbol{n}+\Delta\boldsymbol{n}|p)}{\sum\limits_{\Delta\boldsymbol{n}\in\mathcal{W}^k(p)} w^k(\Delta\boldsymbol{n}|p)\left(|y_A^k(\boldsymbol{n}+\Delta\boldsymbol{n}|p)|^2 + |y_B^k(\boldsymbol{n}+\Delta\boldsymbol{n}|p)|^2\right)},$$

where $\mathcal{W}^k(p)$ is the window at level k and orientation p, and $w^k(\cdot|p)$ its corresponding weights.

By analyzing the match measure, one can determine where the sources differ and to which extent, and use this information to combine them in an appropriate way.

6.2.6 Combination map

This module performs the actual combination of the MR coefficients of the sources. For simplicity, consider two sources and assume that every composite coefficient is ‘assembled’ from the source coefficients at the corresponding level, band and position. More precisely,

$$y_F^k(\cdot) = C^k\left(y_A^k(\cdot), y_B^k(\cdot), d^k(\cdot)\right),$$

---

[5] The match measure is usually a similarity measure between *two* inputs.

where $C^k\colon \mathbb{R}^3 \mapsto \mathbb{R}$ is the combination map at level k. A simple choice for C^k is a linear mapping, e.g.,

$$C^k(y_1, y_2, \delta) = w_A(\delta)y_1 + w_B(\delta)y_2\,, \tag{6.3}$$

where the weights $w_A(\delta)$, $w_B(\delta)$ depend on the decision parameter δ.

One can also use nonlinear mappings. Some well-known nonlinear mappings are multidimensional scaling [86], Sammon's mapping [126] and self-organizing maps [82]. Such mapping techniques have often been used for visualization of high-dimensional data sets [101].

In the sequel, we restrict ourselves to linear combination maps as in (6.3), yet with possibly more than two input sources. Thus, the composite coefficients $y_F^k(\cdot)$ are obtained by an *additive or weighted combination*:

$$y_F^k(\cdot) = \sum_{S\in\mathcal{S}} w_S\big(d^k(\cdot)\big)y_S^k(\cdot)\,. \tag{6.4}$$

For the particular case where only one of the coefficients $y_S^k(\cdot)$ has a weight distinct from zero, that is, only one of the sources contributes to the composite, we talk about *selective combination* or *combination by selection*.

6.2.7 Decision map

The construction of the decision map is a key point in our approach because its output d^k governs the combination map C^k. Therefore, the decision map actually determines the combination of the various MR decompositions y_S and hence the construction of the composite y_F.

In our case, where we assume a weighted combination such as in (6.4), the decision map controls the values of the weights to be assigned to each of the source coefficients. Indeed, specifying the decision $\delta = d^k(\cdot)$ is, in practice, equivalent to specifying the weights $w_S(\delta)$. For this reason, the combination and decision maps are often 'grouped' together by expressing the composite coefficients in terms of the parameters or functions the decision is based on. The problem of 'how to compute $d^k(\cdot)$' is translated into the problem of 'how to compute $w_S(d^k(\cdot))$'. A natural approach is to assign to each coefficient a weight that depends increasingly on the activity. In general, the resulting weighted average (performed by the combination map) leads to a stabilization of the fusion result, but it introduces the problem of contrast reduction in case of opposite contrast in different source images. This can be avoided by using a selective rule that picks the most salient component, i.e., the one with largest activity. In this case, we get after applying the combination map:

$$y_F^k(\cdot) = y_M^k(\cdot) \qquad \text{with } M = \operatorname{argmax}_S a_S^k(\cdot)\,. \tag{6.5}$$

In other words, the decision map 'decides' that the most salient coefficient (among the various $y_S^k(\cdot)$, $S \in \mathcal{S}$) is the best choice for the composite coefficient $y_F^k(\cdot)$, and 'tells' the combination map to select it, i.e.,

$$w_S(d^k(\cdot)) = \begin{cases} 1 & \text{if } S = \operatorname{argmax}_{S'} a_{S'}^k(\cdot) \\ 0 & \text{otherwise.} \end{cases}$$

This selective combination is also known in the literature as a *'choose max'* selection or *maximum selection* rule. For the case of two input sources, (6.5) can be written as

$$y_F^k(\cdot) = \begin{cases} y_A^k(\cdot) & \text{if } a_A^k(\cdot) > a_B^k(\cdot) \\ y_B^k(\cdot) & \text{otherwise.} \end{cases}$$

It works well under the assumption that at each image location, only one of the source images provides the most useful information. This assumption is not always valid, and a weighted combination may appear a better option. Alternatively, a match measure can be used to decide how to combine the coefficients. For instance,

$$y_F^k(\cdot) = \begin{cases} y_A^k(\cdot) & \text{if } m_{AB}^k(\cdot) \le T \text{ and } a_A^k(\cdot) > a_B^k(\cdot) \\ y_B^k(\cdot) & \text{if } m_{AB}^k(\cdot) \le T \text{ and } a_A^k(\cdot) \le a_B^k(\cdot) \\ \dfrac{y_A^k(\cdot) + y_B^k(\cdot)}{2} & \text{if } m_{AB}^k(\cdot) > T, \end{cases}$$

for some threshold T. Thus, at locations where the source images are distinctly different, the combination process selects the most salient component, while at locations where they are similar, the process averages the source components. In this manner, averaging reduces noise and provides stability at locations where source images contain similar information, whereas selection retains salient information and reduces artifacts due to opposite contrast at locations where both source images are different.

In the examples presented so far, the decision is taken for each coefficient without reference to the others. This may degrade the fusion result since there is the possibility of feature cancellation when the inverse transform is applied to obtain the composite image. One may avoid these problems to some extent by taking into account the spatial, inter- and intra-scale dependencies between the coefficients. Note that by construction, each coefficient of an MR decomposition can be related to a set of coefficients in other orientation bands and other levels: they represent the same (or nearby) spatial location in the original image. It seems reasonable then to consider all (or a set of) these coefficients when determining the composite MR representation. For example, one may use intra-scale dependencies to obtain:

$$y_F^k(\boldsymbol{n}|p) = \begin{cases} y_A^k(\boldsymbol{n}|p) & \text{if } \sum_{p'=1}^{P} a_A^k(\boldsymbol{n}|p') > \sum_{p'=1}^{P} a_B^k(\boldsymbol{n}|p') \\ y_B^k(\boldsymbol{n}|p) & \text{otherwise.} \end{cases}$$

Here, each composite coefficient is obtained by a selective rule which takes into account the corresponding activity values of *all* detail bands p. In this particular example, composite coefficients of different orientation bands (but at the same level and location) are selected from the same source.

Another possibility is to exploit spatial redundancy between neighboring samples. One may assume that neighboring samples are likely to belong to the same object and thus should be treated in the same way. An illustrative example is the consistency verification method proposed by Li *et al.* [92]. This method consists in applying a majority filter to a preliminary decision map. Now the filtered decision map determines the combination of the images. For example, consider the case where, according to the preliminary decision map, the composite $y_F^k(\cdot)$ should be selected from y_A^k, while the majority of the surrounding composite coefficients should be selected from y_B^k. After the filtering, the decision map will indicate that the composite coefficient $y_F^k(\cdot)$ should be selected from y_B^k.

Such modifications of the decision map are motivated by the fact that significant image features tend to be stable with respect to variations in space, scale and orientation. Thus, when comparing the corresponding image features in multiple source images, consideration of the dependencies between transform coefficients may lead to a more robust fusion strategy.

Combination of approximation images vs. combination of detail images

Because of their different physical meaning, the approximation and detail images are usually treated by the combination algorithm in a different fashion. For the detail images y_S^k, one may observe that relevant perceptual information relates to the 'edge' information that is present in each of the detail coefficients $y_S^k(\cdot)$. Detail coefficients having large absolute values correspond to sharp intensity changes and hence to salient features in the image such as edges, lines and region boundaries. The nature of the approximation coefficients, however, is different. The approximation image $y_S^K(\cdot|0)$ represents a coarse representation of the original image x_S and may have inherited some of its properties such as the mean intensity or some coarse texture information. Thus, coefficients $y_S^K(\boldsymbol{n}|0)$ with high magnitudes do not necessarily correspond with salient features. Therefore, an activity measure $a_S^K(\cdot|0)$ based on quantities such as entropy, variance or texture criteria, is more appropriate than one based on energy like in (6.2).

In many approaches, the composite approximation coefficients of the highest decomposition level, representing the mean intensity, are taken to be a weighted average of the approximation of the sources:

$$y_F^K(\boldsymbol{n}|0) = \frac{\sum_{S\in\mathcal{S}} y_S^K(\boldsymbol{n}|0)}{|S|}, \tag{6.6}$$

where $|S|$ is the number of sources. The logic behind this combination relies on the assumptions that the sources x_S are contaminated by additive Gaussian noise and that, provided that K is high enough, the relevant features have already been captured by the details $y_S^k(\cdot|p)$. Thus, the approximation images $y_S^K(\cdot|0)$ of the various sources contain the same scene distorted by noise; averaging them reduces the variance of the noise while ensuring that an appropriate mean intensity is maintained.

A popular way to construct the composite y_F is to use (6.6) for the approximation coefficients and the selective combination in (6.5) for the detail coefficients. For the simple case where

$a_S^k(\cdot) = |y_S^k(\cdot)|$, and there are two input sources, we can express this combination algorithm as

$$y_F^K(\boldsymbol{n}|0) = \frac{y_A^K(\boldsymbol{n}|0) + y_B^K(\boldsymbol{n}|0)}{2} \tag{6.7}$$

$$y_F^k(\boldsymbol{n}|p) = \begin{cases} y_A^k(\boldsymbol{n}|p) & \text{if } |y_A^k(\boldsymbol{n}|p)| > |y_B^k(\boldsymbol{n}|p)| \\ y_B^k(\boldsymbol{n}|p) & \text{otherwise.} \end{cases} \quad p = 1, \ldots, P \tag{6.8}$$

Note that other factors may be incorporated for the fusion rules. In particular, if some prior knowledge is available, all the fusion blocks can use such information to improve fusion performance. For instance, when combining the source coefficients, the weights assigned to them may depend not only on the activity and match measure, but may also reflect some a priori knowledge of a specific type, giving preference to certain levels k, bands p, locations $\boldsymbol{n}$, or some input sources S.

Finally, we want to remark that the decision on which techniques to use is very much driven by the application. At the same time, the characteristics of the resulting composite image depend strongly on the applied pre-processing and the chosen fusion techniques. The different options we have presented are neither exhaustive nor mutually exclusive and they should merely be considered as practically important examples.

6.3 Overview of some existing fusion schemes

In the literature one finds several MR fusion approaches which fit into our general scheme. In this section, we review some of them. The reader may also get an impression of the evolution of MR fusion schemes during the past fifteen years.

The first MR image fusion approach proposed in the literature is due to Burt [17]. His implementation used the Laplacian pyramid (see Section 1.3) and the sample-based maximum selection rule with $a_S^k(\cdot) = |y_S^k(\cdot)|$. Thus, each composite coefficient $y_F^k(\cdot)$ is obtained by

$$y_F^k(\cdot) = \begin{cases} y_A^k(\cdot) & \text{if } |y_A^k(\cdot)| > |y_B^k(\cdot)| \\ y_B^k(\cdot) & \text{otherwise.} \end{cases} \tag{6.9}$$

Toet [152, 155] presented a similar algorithm but using the ratio-of-low-pass pyramid. His approach is motivated by the fact that the human visual system is based on contrast, and therefore, a fusion technique which selects the highest local luminance contrast is likely to provide better details to a human observer. Another variation of this scheme is obtained by replacing the linear filters by morphological ones [103, 153].

Burt and Kolczynski [21] proposed to use the gradient pyramid, such as described in Section 2.5.5, together with a combination algorithm that is based on an activity and a match measure. In particular, they define the activity of $y_S^k(\cdot)$ as a local energy measure:

$$a_S^k(\boldsymbol{n}|p) = \sum_{\Delta\boldsymbol{n} \in \mathcal{W}^k(p)} \left|y_S^k(\boldsymbol{n} + \Delta\boldsymbol{n}|p)\right|^2, \tag{6.10}$$

and the match between $y_A^k(\cdot)$ and $y_B^k(\cdot)$ as

$$m_{AB}^k(\boldsymbol{n}|p) = \frac{2\sum_{\Delta\boldsymbol{n}\in\mathcal{W}^k(p)} y_A^k(\boldsymbol{n}+\Delta\boldsymbol{n}|p)y_B^k(\boldsymbol{n}+\Delta\boldsymbol{n}|p)}{a_A^k(\boldsymbol{n}|p)+a_B^k(\boldsymbol{n}|p)}\,, \tag{6.11}$$

with $\mathcal{W}^k(p)$ being either a 1×1, 3×3, or 5×5 window centered at the origin. The combination process is the weighted average

$$y_F^k(\cdot) = w_A\big(d^k(\cdot)\big)y_A^k(\cdot) + w_B\big(d^k(\cdot)\big)y_B^k(\cdot)\,,$$

where the weights are determined by the decision process for each level k, band p, and location $\boldsymbol{n}$ as $w_A\big(d^k(\cdot)\big) = 1 - w_B\big(d^k(\cdot)\big) = d^k(\cdot)$, with

$$d^k(\cdot) = \begin{cases} 1 & \text{if } m_{AB}^k(\cdot) \le T \text{ and } a_A^k(\cdot) > a_B^k(\cdot) \\ 0 & \text{if } m_{AB}^k(\cdot) \le T \text{ and } a_A^k(\cdot) \le a_B^k(\cdot) \\ \frac{1}{2}+\frac{1}{2}(\frac{1-m_{AB}^k(\cdot)}{1-T}) & \text{if } m_{AB}^k(\cdot) > T \text{ and } a_A^k(\cdot) > a_B^k(\cdot) \\ \frac{1}{2}-\frac{1}{2}(\frac{1-m_{AB}^k(\cdot)}{1-T}) & \text{if } m_{AB}^k(\cdot) > T \text{ and } a_A^k(\cdot) \le a_B^k(\cdot) \end{cases} \tag{6.12}$$

for some threshold T. Observe that in case of a poor match (no similarity between the inputs), the source coefficient having the largest activity will yield the composite value, while otherwise, a weighted sum of the sources coefficients will be used. The authors claim that this approach provides a partial solution to the problem of combining components that have opposite contrast, since such components are combined by selection. In addition, the use of area-based (vs. sampled-based) operations and the gradient pyramid provide greater stability in noise, compared to the Laplacian pyramid-based fusion.

Ranchin and Wald [119] presented one of the first wavelet-based fusion systems. This approach is also used by Li *et al.* in [92]. Their implementation considers the maximum absolute value within a window as the activity measure associated with the sample centered in the window:

$$a_S^k(\boldsymbol{n}|p) = \max_{\Delta\boldsymbol{n}\in\mathcal{W}^k(p)}\big|y_S^k(\boldsymbol{n}+\Delta\boldsymbol{n}|p)\big|\,.$$

For each position in the transform domain, the maximum selection rule is used to determine which of the inputs is likely to contain the most useful information. This results in a preliminary decision map which indicates, at each position, which source should be used in the combination map. This decision map is then subject to a consistency verification. In particular, Li *et al.* apply a majority filter in order to remove possible wrong selection decisions caused by impulsive noise. The authors claim that their scheme performs better than the Laplacian pyramid-based fusion due to the compactness, directional selectivity and orthogonality of the discrete wavelet transform (DWT).

Wilson *et al.* [169] used a DWT fusion method and a perceptual-based weighting based on the frequency response of the human visual system. In fact, their activity measure is computed as a weighted sum of the Fourier transform coefficients of the wavelet decomposition, with the weights determined by the contrast sensitivity[6]. They define a perceptual distance between the sources as

$$D_{AB}^k(\cdot) = \left| \frac{a_A^k(\cdot) - a_B^k(\cdot)}{a_A^k(\cdot) + a_B^k(\cdot)} \right| ,$$

and use this together with the activity to determine the weights of the wavelet coefficients from each source. Observe that this perceptual distance is directly related to the matching measure: the smaller the perceptual distance, the higher the matching measure. The final weighting is given by $w_A\big(d^k(\cdot)\big) = 1 - w_B\big(d^k(\cdot)\big) = d^k(\cdot)$, with:

$$d^k(\cdot) = \begin{cases} 1 & \text{if } D_{AB}^k(\cdot) > T \text{ and } a_A^k(\cdot) > a_B^k(\cdot) \\ 0 & \text{if } D_{AB}^k(\cdot) > T \text{ and } a_A^k(\cdot) \leq a_B^k(\cdot) \\ 1 - \frac{1}{2}\frac{a_B^k(\cdot)}{a_A^k(\cdot)} & \text{if } D_{AB}^k(\cdot) \leq T, \end{cases}$$

for some threshold T. In the experimental results presented by the authors, the composite images obtained with their method are visually better than the ones obtained by fusion techniques based on the gradient pyramid or the ratio-of-low-pass pyramid.

Koren *et al.* [83] used a steerable pyramid transform (see Section 2.5.4) for the MR decomposition. They advocate their choice because of the shift-invariance and non-aliasing properties this transform offers. For each orientation band, the activity is a local oriented energy. Only the components corresponding to the orientation band whose activity is the largest are included for reconstruction (maximum selection rule). Liu *et al.* [94] take a completely different point of view. They also used a steerable pyramid but rather than using it to fuse the source images, they fuse the various bands of this decomposition by means of a Laplacian pyramid.

In [123], Rockinger considered an approach based on a shift-invariant extension of the DWT. The detail coefficients are combined by a maximum selection rule, while the coarse approximation coefficients are merged by averaging. Due to the shift-invariance representation, the proposed method is particularly useful for image sequence fusion, where a composite image sequence has to be built from various input image sequences. The author shows that the shift-invariant fusion method outperforms other MR fusion methods with respect to temporal stability[7] and consistency[8].

Pu and Ni [116] proposed a contrast-based image fusion method using the DWT. They measure the activity as the absolute value of what they call directive contrast:

$$a_S^k(\boldsymbol{n}|p) = \left| \frac{y_S^k(\boldsymbol{n}|p)}{y_S^k(\boldsymbol{n}|0)} \right| \qquad p = 1, \ldots, 3 ,$$

---

[6] The contrast sensitivity is defined as the reciprocal of the lowest contrast value above which a given spatial frequency is perceived.

[7] A composite image sequence is temporally stable if the graylevel changes in the composite sequence are only caused by the graylevel changes in the input sequences.

[8] A composite image sequence is temporally consistent if the graylevel changes occurring in the input sequences are present in the composite sequence without any delay or contrast change.

and use a maximum selection rule as the combination method of the wavelet coefficients. They also proposed an alternative approach where the combination process is performed on the directive contrast itself.

Li and Wang [93] examined the application of discrete multiwavelet transforms (see Section 2.5.3) to multisensor image fusion. The composite coefficients are obtained through a sample-based maximum selection rule. The authors showed experimental results where their fusion scheme performs better than those based on comparable scalar wavelet transforms.

Another MR technique is proposed by Scheunders in [128] where the fusion consists of retaining the modulus maxima [100] of the wavelet coefficients from the different bands and combining them. Noise reduction can be applied during the fusion process by removing noise-related modulus maxima. In the experiments presented, the proposed method outperforms other wavelet-based fusion techniques. A different yet interesting approach by Scheunders [129, 130] uses a wavelet representation based on multiscale fundamental forms.

Mukhopadhyay and Chanda [111] presented a fusion scheme using multiscale morphology. In particular, they employ two MR top-hat transforms [131,140] for extracting bright and dark details from the sources. For each source x_S, they derive two MR structures $y_{S,b}$ and $y_{S,d}$, by applying the bright and dark top-hat transforms respectively. A sample-based maximum selection rule on all $y_{S,b}$ yields a 'bright composite' $y_{F,b}$. Likewise, the same selection rule on all $y_{S,d}$ produces a 'dark composite' $y_{F,d}$. In both cases, the activity measure of each sample is taken to be its amplitude. Finally, the two composite MR representations are combined by subtracting the 'dark' details $y^k_{F,d}$ from the 'bright' ones $y^k_{F,b}$ and summing up all the entries. The 'dark' and 'bright' approximations are averaged and added to the detail to obtain an output composite image x_F.

6.4 Examples

Fig. 6.2-Fig. 6.5 are examples of composite images obtained by choosing different alternatives in the fusion blocks. Here, we give a few more examples. In all cases, we have used the sources shown in Fig. 6.1, which correspond to visual and IR image modalities. For displaying purposes the gray values of the pixels have been scaled between 0 and 255 (histogram stretching). Unless otherwise stated, three levels of decomposition (i.e., $K=3$) have been used for the MR decomposition of the sources.

Fig. 6.8 shows examples of composite images obtained by some of the fusion algorithms which have been discussed before.

The first row of Fig. 6.8 corresponds to the special case where $K = 0$ and thus no MR decomposition is done. Fig. 6.8(a) is the result of a pixel-by-pixel average of the sources, while Fig. 6.8(b) is a weighted average where the weights have been determined by a principal component analysis (PCA). In the average-fused image, we can observe the loss of contrast compared to the other examples. The PCA-fused image strongly resembles the visual image (Fig. 6.1(a)) and the person is almost invisible.

In Fig. 6.8(c) we have used Burt's method [17] comprising a Laplacian pyramid decomposition and the combination algorithm specified in (6.7)-(6.8). The same combination algorithm but using a ratio-of-low pass pyramid for the decomposition (Toet's method [152]) yields the result in Fig. 6.8(d).

Fig. 6.8(e) illustrates the fusion algorithm proposed by Burt and Kolczynski in [21]. Here, the activity measure, the match measure, and the weights are computed as in (6.10)-(6.12), with a 3×3 window centered at the origin and a threshold $T = 0.85$.

Fig. 6.8(d) shows the composite image obtained by the fusion scheme proposed by Li *et al.* in [92]. We have used the Daubechies orthogonal wavelet of order 2 to compute the DWT, and a 3×3 window for the consistency verification.

Fig. 6.9 shows some examples obtained by combining different alternatives in the different fusion blocks. The purpose is not to outperform the already existing approaches but to give an idea of the flexibility and freedom our framework offers.

In Fig. 6.9(a) a steerable pyramid with $P=4$ is used for the decomposition of the sources. The combination algorithm is the same as the one proposed by Burt and Kolczynski, with a 3×3 window centered at the origin and a threshold $T = 0.85$. Fig. 6.9(b) has also been obtained with the same combination algorithm but performing, in addition, consistency verification, and using a translation invariant Haar wavelet for the MR representation of the sources.

Fig. 6.9(c)-(d) are examples of composite images where a median pyramid is employed for the MR decomposition. In Fig. 6.9(d), however, the ratio of the median filtered approximations instead of the standard difference is used. In both cases, the simple combination algorithm specified in (6.7)-(6.8) is applied.

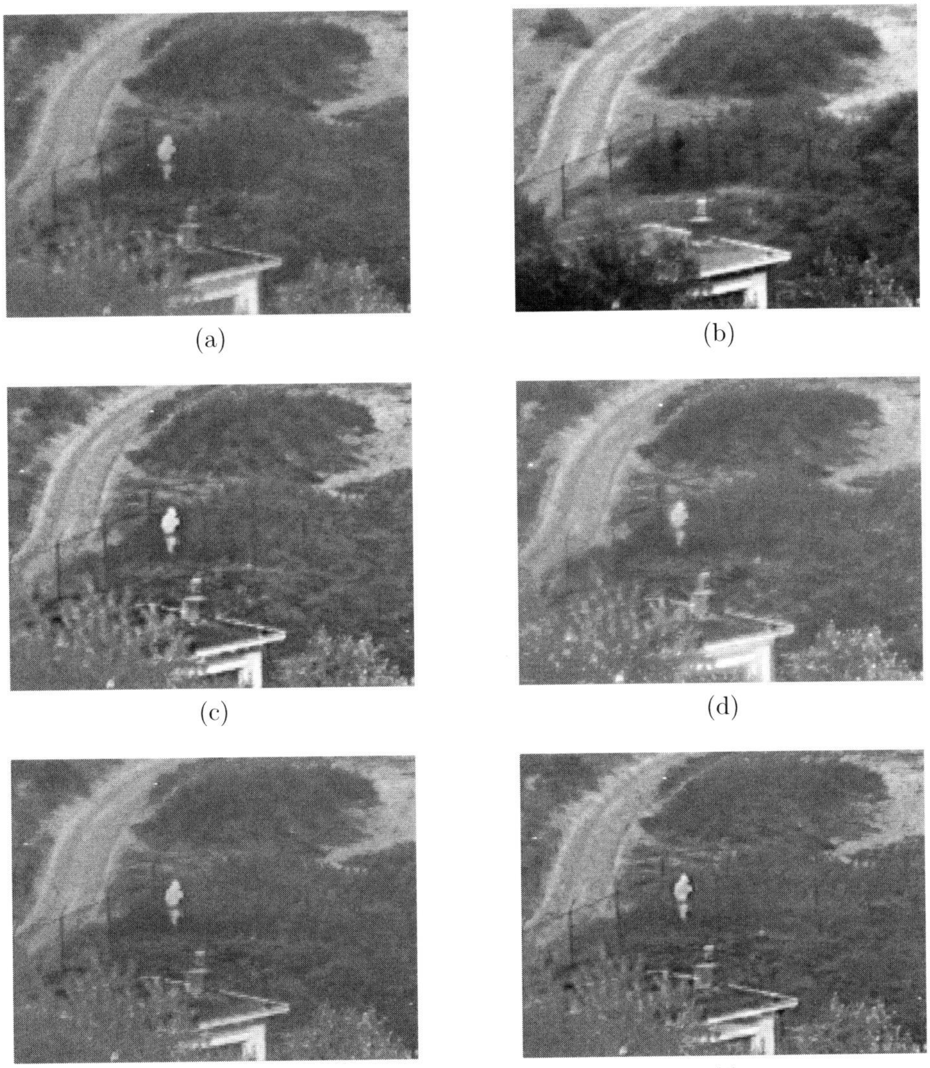
In Fig. 6.9(e)-Fig. 6.9(f), the lifting scheme (see Section 2.2) is used to perform the MR decomposition on a quincunx lattice. Fig. 6.9(e) shows the composite image obtained with the max-lifting scheme[9] [70], while Fig. 6.9(f) depicts the result obtained with a lifting scheme where the prediction and update operators are Neville filters [84] of order 2. In both cases the combination algorithm specified in (6.7)-(6.8) is used.

6.5 Adaptive MR schemes for image fusion

The use of adaptive MR transforms opens a new perspective on MR fusion algorithms. Note that classical MR fusion approaches simply apply a *fixed* transform to the sources x_S, combine the resulting MR coefficients into a composite MR decomposition y_F, and apply the inverse of the fixed transform to y_F. If we use an adaptive transform, the decomposition is steered by the input data which, in effect, results in different transforms for different sources x_S. We are then immediately led to the problem of which inverse transform to apply to the combined data y_F. To get around this problem, we look for a joint MR transform which is adapted to all sources x_S. This *joint adaptive MR transform* is used to decompose all x_S, and its inverse can be used to compute the composite image x_F from the composite MR decomposition y_F. The

---

[9] The max-lifting scheme is based on morphological operators. A major characteristic of this scheme is that it preserves local maxima.

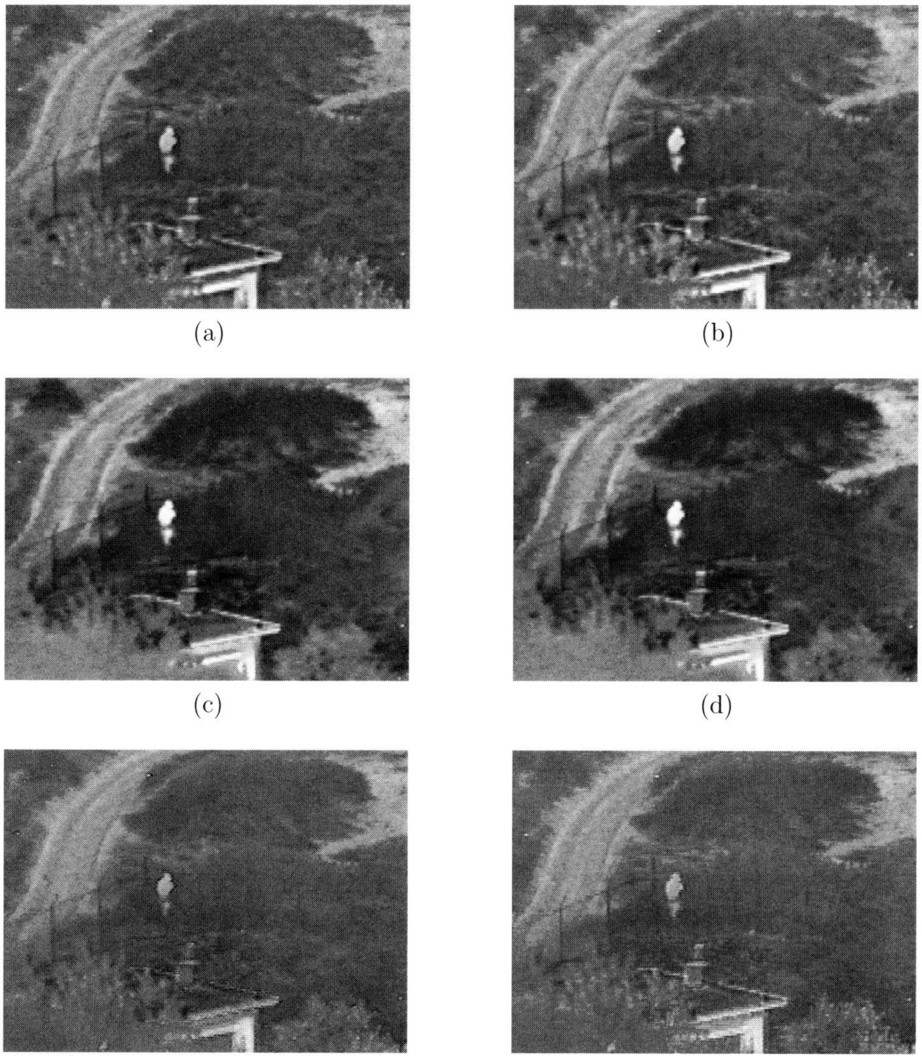


(a) (b)

(c) (d)

(e) (f)

Figure 6.8: *Examples of fusion by some existing methods: (a) average; (b) weighted average by PCA; (c) Burt's method; (d) Toet's method; (e) Burt and Kolczynski's method; (f) Li et al.'s method.*



(a) (b)

(c) (d)

(e) (f)

Figure 6.9: *Examples of fusion by other methods: (a) steerable pyramid with Burt and Kolczynski's combination algorithm; (b) undecimated DWT with Burt and Kolczynski's combination algorithm and consistency verification. For (c)-(f), the combination algorithm is* (6.7)-(6.8) *and the MR decompositions are: (c) median pyramid; (d) ratio-of-median pyramid; (e) max-lifting wavelet scheme (in quincunx lattice); (f) linear lifting with Neville filters wavelet scheme (in quincunx lattice).*

challenge therefore is to find joint adaptive MR transforms that are suited for the purpose of image fusion.

In this section we investigate some simple approaches. In all cases, we consider two input sources x_A, x_B, and assume lifting-based wavelet decompositions (see below) in which the choice of the update lifting step is governed by some decision map[10]. The general idea is the following:

1. choose a criterion for triggering the update lifting step;
2. compute the update decision map for each input;
3. merge the update decision maps into a *joint update decision map*;
4. compute the transform on each input by using the joint decision map.

This procedure is iterated over the approximation images yielding two MR decompositions y_A and y_B, and the multiscale joint decision map

$$D_F = \{D_F^1, \ldots, D_F^K\}$$

where D_F^k is the joint decision map used at level k.

Lifting-based wavelet decompositions

We consider MR decompositions based on the lifting scheme; see Chapters 3–5 for more details. First, an original image x is split into its four polyphase components: $x_0^1(\boldsymbol{n}) = x(2m, 2n)$, $y_0^1(\boldsymbol{n}|1) = x(2m, 2n+1)$, $y_0^1(\boldsymbol{n}|2) = x(2m+1, 2n)$, $y_0^1(\boldsymbol{n}|3) = x(2m+1, 2n+1)$. Then, x_0^1 is updated by

$$x^1(\boldsymbol{n}) = x_0^1(\boldsymbol{n}) \oplus_{d_{\boldsymbol{n}}} U_{d_{\boldsymbol{n}}}(y_0^1)(\boldsymbol{n}),$$

where $\oplus_{d_{\boldsymbol{n}}}$, $U_{d_{\boldsymbol{n}}}$ depend on $d_{\boldsymbol{n}}$ which is the output of some decision map D^1 at location $\boldsymbol{n}$, i.e., $d_{\boldsymbol{n}} = D^1(\boldsymbol{n})$. The polyphase components $y_0^1(\boldsymbol{n}|p)$, $p = 1, \ldots, 3$, are predicted as in (4.16), i.e., $y^1(\boldsymbol{n}|p) = y_0^1(\boldsymbol{n}|p) - x^1(\boldsymbol{n})$ for $p = 1, 2$, and $y^1(\boldsymbol{n}|3) = y_0^1(\boldsymbol{n}|3) - x^1(\boldsymbol{n}) - y^1(\boldsymbol{n}|1) - y^1(\boldsymbol{n}|2)$. The component x^1 is the approximation image at level 1 and $y^1 = \{y^1(\cdot|1), y^1(\cdot|2), y^1(\cdot|3)\}$ are the detail images at level 1. The wavelet step (splitting, update and prediction) is iterated over the approximation image until level K is reached. Note that at each level k the update lifting step depends on D^k while the prediction lifting step is fixed.

6.5.1 Case studies

In the examples presented below, we obtain the composite detail coefficients by a sample-based maximum selection rule as in (6.9). The MR decomposition y_F is inverted by reverting the lifting steps described above. In particular, the inversion of the update lifting step is given by

$$x_{0,F}^k(\boldsymbol{n}) = x_F^k(\boldsymbol{n}) \ominus_{d_{\boldsymbol{n}}} U_{d_{\boldsymbol{n}}}(y_{0,F}^k)(\boldsymbol{n}),$$

---

[10] This decision map should not be confused with the decision map of the MR fusion scheme. In order to avoid confusion, we refer to the former as the *update decision map*, and to the latter as the *fusion decision map*.

where $d_{\boldsymbol{n}} = D_F^k(\boldsymbol{n})$ and $\ominus_d$ denotes the subtraction which inverts $\oplus_d$. The composite image is $x_F = x_F^0$, obtained from merging $x_{0,F}^0$ with $y_{0,F}^0$.

In the two first simulations, D_A, D_B are computed as described in Experiment 4.2.5, i.e.,

$$D_S^k(\boldsymbol{n}) = [p(\boldsymbol{v}_S^k(\boldsymbol{n})) > T_S]\,, \quad S = A, B\,, \tag{6.13}$$

where $\boldsymbol{v}_S^k$ is the gradient vector at level k and p is the weighted Euclidean norm:

$$p(\boldsymbol{v}) = \left(\sum_{j=1}^{4} |v_j|^2 + \frac{1}{2}(\sum_{j=5}^{8} |v_j|^2)\right)^{1/2} .$$

The filters used in the update lifting step are also the same as in Experiment 4.2.5 (i.e., a weighted average for $d = 0$ and the identity filter for $d = 1$). Note that these filters are chosen in such a way that the adaptive scheme satisfies the threshold criterion (see Proposition 4.2.5). However, since for the reconstruction we use a given decision map, we could have chosen any other filters. Recall that in previous chapters, the threshold criterion was important in order to ensure perfect reconstruction without the need for bookkeeping. In this section, we do store the decisions we make for the update filters. This bookkeeping (as long as we are not concern about memory storage in the system) is not a serious issue in image fusion, however.

Case 1: D_F as the 'union' of D_A, D_B - Fig. 6.10
In this example, we consider the fusion of a magnetic resonance image (MRI) and a computer tomography (CT) image; see top row of Fig. 6.10. At each level k, we obtain their respective update decision maps D_A^k, D_B^k as described above, using $T_A = 40$, $T_B = 60$. We have used different thresholds because of the difference statistics of the images. Then, we construct the joint decision map D_F^k by

$$D_F^k(\boldsymbol{n}) = \max\{D_A^k(\boldsymbol{n}), D_B^k(\boldsymbol{n})\}\,.$$

Since $D_S^k(\boldsymbol{n}) \in \{0, 1\}$, the previous expression means that D_F^k is obtained as the 'union' of D_A^k and D_B^k.

The middle row of Fig. 6.10 shows the composite image and the joint update decision maps D_F^1, D_F^2 and D_F^3. The bottom row shows the composite images obtained by the corresponding non-adaptive decompositions with fixed $d = 0$ (left) and $d = 1$ (right).

We can observe that the composite image obtained by the adaptive scheme offers a good compromise compared to its non-adaptive counterparts. It reduces both the blurring and ringing of the non-adaptive composite image with $d = 0$, and the blocking artifacts of the non-adaptive composite image with $d = 1$.

Case 2: D_F as the 'intersection' of D_A, D_B - Fig. 6.11
Here we take as sources the images shown at the top row of Fig. 6.11. We obtain D_A^k, D_B^k in the same way as before but with $T_A = T_B = 40$. We construct the joint update decision map by

$$D_F^k(\boldsymbol{n}) = \min\{D_A^k(\boldsymbol{n}), D_B^k(\boldsymbol{n})\}\,.$$

That is, D_F^k is the 'intersection' of the individual update decision maps. This implies that only if both input images have a large gradient no filtering takes place, i.e., $x_S^k(\boldsymbol{n}) = x_S^{k-1}(2\boldsymbol{n})$.

The results are shown in Fig. 6.11. Note that the composite image obtained with the adaptive approach (left image of the second row) is similar to the one obtained with the non-adaptive decomposition with $d = 0$ (left image of the bottom row). Indeed, one can foresee this similarity by inspecting the joint update decision maps which, especially for finer levels, give $d = 0$ for most locations. The smoothing effect of the filter corresponding to $d = 0$ reduces the aliasing and the resulting blocking artifacts that one can perceive in the composite image obtained with fixed $d = 1$ (right image of the bottom row). On the other hand, this latter composite image presents less ringing artifacts than the one obtained with fixed $d = 0$. This ringing is most noticeable in the horizontal edge just above the digit '12'. Since this edge is 'strong' for both input images, it is preserved by the joint adaptive decomposition. Therefore, the composite image obtained with the adaptive approach avoids the ringing as well as the blocking artifacts.

Case 3: combining linear and morphological decompositions - Fig. 6.12
In this example the update decision map alternates between two different filters: an average filter for texture regions ($d = 0$) and a median-based[11] filter for all other regions ($d = 1$). The choice of the filter is made according to the texture criterion proposed in [55]. This criterion is based on the premise that texture regions have a high local variance in all directions.

The composite image obtained using this approach is shown at the left of the middle row of Fig. 6.12. Here, we have considered two levels of decompositions. The non-adaptive counterparts are shown in the bottom row. Observe that, again, with the adaptive approach, both ringing and blocking artifacts have considerably decreased when compared to the images obtained with the non-adaptive approaches.

6.5.2 Discussion

These preliminary experiments show that the use of adaptive transforms may result in a slight reduction of artifacts when compared to their non-adaptive counterparts. However, a thorough investigation is needed in order to find decompositions which can outperform standard transforms for image fusion such as the Laplacian pyramid.

Obviously, the construction of the joint adaptive decomposition is a key factor which needs further research. Currently we construct the joint update decision map from the independent update decision maps of the different sources. However, we can also think of computing such joint decision map by simultaneously analyzing the sources. The choice of the filters is another issue that needs further exploration.

---

[11] The update of a sample is computed by averaging the sample with the median of its eight neighbors.

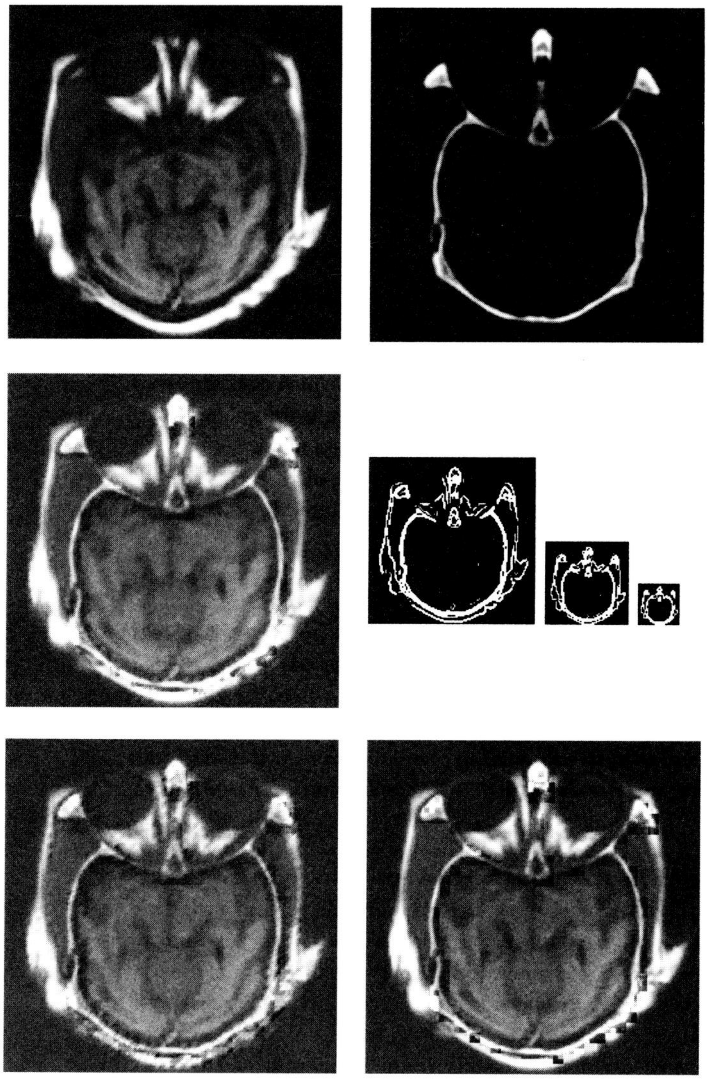


Figure 6.10: *Case* 1. *Top: MRI (left) and CT (right) input images. Middle: composite image in the adaptive case (left) and joint update decision maps (right)* D_F^1, D_F^2, D_F^3. *Bottom: composite images in the non-adaptive cases with* $d = 0$ *(left) and* $d = 1$ *(right).*

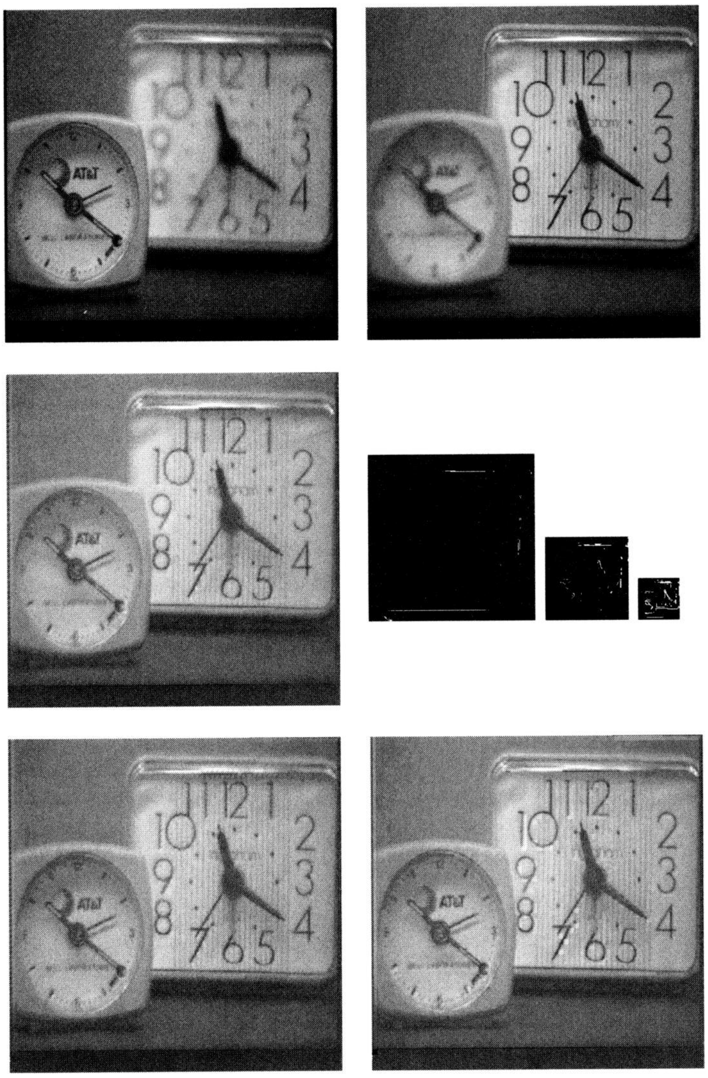


Figure 6.11: *Case 2. Top: out-of-focus input images. Middle: composite image in the adaptive case (left) and joint update decision maps (right)* D_F^1, D_F^2, D_F^3*. Bottom: composite images in the non-adaptive cases with* $d = 0$ *(left) and* $d = 1$ *(right).*

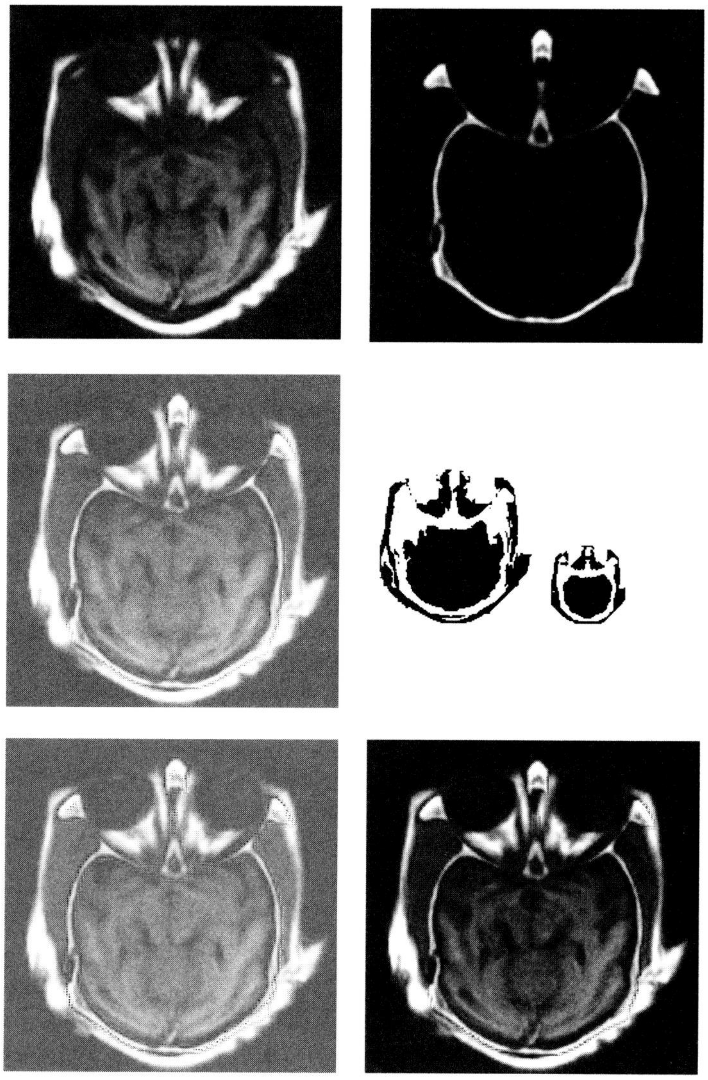


Figure 6.12: *Case 3. Top: MRI (left) and CT (right) input images. Middle: composite image in the adaptive case (left) and joint update decision maps (right)* D_F^1, D_F^2. *Bottom: composite image in the non-adaptive cases with* $d = 0$ *(left) and* $d = 1$ *(right). Here* $d = 0$ *corresponds to an average filter, while* $d = 1$ *to a median-based filter.*

Chapter 7

Region-based multiresolution image fusion

The algorithms based on multiresolution (MR) techniques that we have discussed in the previous chapter are mainly pixel-based approaches where each individual coefficient of the MR decomposition (or possibly the coefficients in a small fixed window) is treated more or less independently. However, for most, if not all, image fusion applications, it seems more meaningful to combine objects rather than pixels. For example, in the input images depicted in Fig. 6.1, a composite image containing objects such as the house, the bushes, the hills, etc., as well as the person from the IR source and the fence from the visual source, would represent a rather accurate description of the underlying scene. Therefore, when fusing these images, it is reasonable to consider the pixels which constitute these objects as entities instead of combining the pixels without reference to the object they belong to. As an intermediate step from pixel-based toward object-based fusion schemes, one may consider region-based approaches. Such approaches have the additional advantage that the fusion process becomes more robust and may help to circumvent some of the well-known drawbacks of pixel-based techniques, such as blurring effects and high sensitivity to noise and misregistration.

In this chapter, we introduce a new region-based approach to MR fusion which combines aspects of feature and pixel fusion. The basic idea is to build a segmentation based on all different source images and to use this segmentation to guide the combination process. A major difference with other existing region-based approaches [105,174] is that the segmentation performed is: (*i*) *multisource*, in the sense that a *single* segmentation is obtained from all the input images, and (*ii*) *multiresolution*, in the sense that it is computed in an MR fashion (thus, we do not merely compute independent segmentations of images at different resolutions). For instance, in [174] the regions are obtained by segmenting (independently) each of the approximation images x_S^K and by exploiting the tree-structure in the decomposition: every detail coefficient $y_S^k(\cdot)$ is assigned to a region in x_S^K.

7.1 The overall scheme: from pixels to regions

7.1.1 Introduction

Our region-based fusion scheme depicted in Fig. 7.1 extends the pixel-based fusion approach discussed in Chapter 6 (see Fig. 6.7). Indeed, it includes all the blocks described before. The major difference between the two schemes consists hereof that the region-based scheme also contains a segmentation module which uses all sources x_S as input and returns a single MR segmentation $\mathcal{R}$ (i.e., a partition of the underlying image domains into regions) as output. Thus, we use MR decompositions to represent the input images at different scales and, additionally, we introduce a multiresolution/multisource (MR/MS) segmentation to partition the image domain at these scales into regions. The activity and match measures are now computed for every such region. These measures may correspond to low-level as well as intermediate-level structures. Furthermore, the MR segmentation $\mathcal{R}$ allows us to impose data-dependent consistency constraints based on spatial as well as inter- and intra-scale dependencies. All this information, i.e., the measures and the consistency constraints, is integrated to yield a decision map d which governs the combination of the coefficients of the sources. This combination results in an MR decomposition y_F, and by MR synthesis we obtain a composite image x_F.

The main functional blocks of this fusion strategy are depicted in Fig. 7.1. Since we already discussed most of them in Chapter 6, we concentrate on the segmentation module and its interaction with the other modules.

7.1.2 MR/MS segmentation

The MR/MS segmentation uses the various source images as input and returns a single MR segmentation

$$\mathcal{R} = \{\mathcal{R}^1, \mathcal{R}^2, \ldots, \mathcal{R}^K\}$$

as output. Here $\mathcal{R}^k$ represents a segmentation at level k, i.e., a partitioning of the domain at level k.

Loosely speaking, $\mathcal{R}$ provides an MR representation of the various regions of the underlying scene. This representation will guide the other blocks of the fusion process; hence instead of working at pixel level, they will take into consideration the regions inferred by the segmentation. From an intuitive point of view, we can regard these regions as the constituent parts of the objects in the overall scene.

In our image fusion problem, segmentation is merely a preparatory step toward actual fusion. In fact, we are not interested in the segmentation of the images *per se*, but rather in a coarse partition of the underlying scene. Therefore, the segmentation process does not need to be extremely accurate. For our purposes, we have developed an MR/MS segmentation algorithm based on *pyramid linking* [20]. We describe our segmentation algorithm in Section 7.2. Obviously, other segmentation methods can be used, as long as they meet the constraint that the sampling structure in $\mathcal{R}$ is the same as in y_S, so that each partition $\mathcal{R}^k$ corresponds to a partition of the detail image $y_S^k(\cdot|p)$.

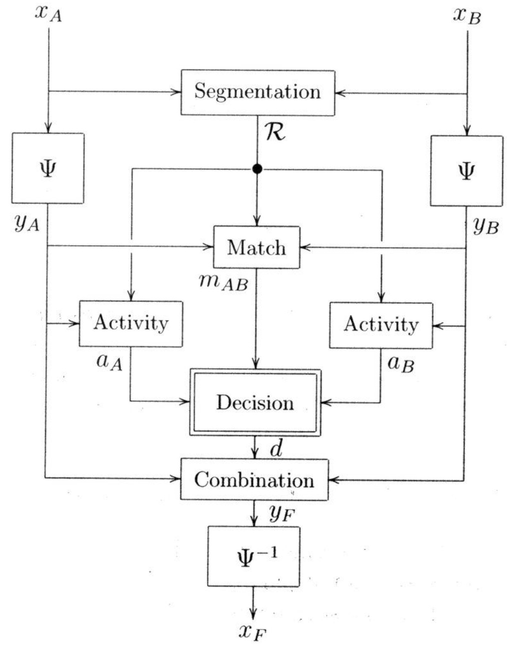


Figure 7.1: *Generic region-based MR fusion scheme with two input sources x_A and x_B, and one output composite image x_F.*

7.1.3 Combination algorithm

Since the building blocks of the combination algorithm in the region-based approach are essentially the same as in the pixel-based case, the combination algorithms discussed in Chapter 6 can be easily extended to the region-based approach. For example, we can define the activity of each region $R \in \mathcal{R}^k$ in $y_S^k(\cdot|p)$ by

$$a_S^k(R|p) = \frac{1}{|R|} \sum_{\boldsymbol{n} \in R} a_S^k(\boldsymbol{n}|p)\,, \tag{7.1}$$

where $|R|$ is the area of region R. Similarly, we can define the match measure of each region $R \in \mathcal{R}^k$ in the image bands $y_A^k(\cdot|p)$ and $y_B^k(\cdot|p)$ by

$$m_{AB}^k(R|p) = \frac{1}{|R|} \sum_{\boldsymbol{n} \in R} m_{AB}^k(\boldsymbol{n}|p)\,. \tag{7.2}$$

Given these measures, the decision map can be constructed in several ways as discussed in Section 6.2.7, with the only difference that $a_S^k(R|p)$, $m_{AB}^k(R|p)$ are used instead of $a_S^k(\boldsymbol{n}|p)$,

$m^k_{AB}(\boldsymbol{n}|p)$. For instance, a combination algorithm based on a maximum selection rule (see (6.5) for the pixel-based case) would read:

$$y^k_F(\boldsymbol{n}|p) = \begin{cases} y^k_A(\boldsymbol{n}|p) & \text{if } a^k_A(R|p) > a^k_B(R|p) \\ y^k_B(\boldsymbol{n}|p) & \text{otherwise} \end{cases} \qquad \text{for all } \boldsymbol{n} \in R. \tag{7.3}$$

As in the pixel-based scheme, once the decision map is constructed, the mapping performed by the combination process is determined for all coefficients, and the synthesis process yields the composite image x_F.

Note that for the particular case in which each region corresponds to a single point $\boldsymbol{n}$, the region-based approach reduces to a pixel-based approach. Thus, the region-based MR fusion scheme extends and generalizes the pixel-based approach, and offers a general framework for MR-based image fusion which encompasses most of the existing MR fusion algorithms.

7.2 MR/MS segmentation based on pyramid linking

In this section we present an MR/MS segmentation algorithm based on pyramid linking. We first review the basics of the conventional pyramid linking segmentation method. Then, we modify and extend this method for the segmentation of several input images.

7.2.1 The linked pyramid

The linked pyramid structure was first described by Burt *et al.* [20] (related work can be found in [9, 28, 75, 113, 164]). It consists of an MR decomposition of an image with the bottom level containing the full-resolution image, and each successive higher level containing a filtered and subsampled image derived from the level below it. The various levels of the pyramid are 'linked' by means of so-called child-parent relations (see Fig. 7.2) between their samples (pixels); such child-parent links are established during an iterative processing procedure to be described below.

A conventional linked pyramid is constructed as follows. First, an approximation pyramid (see Section 2.3) is produced by low-pass filtering and sampling. Then, child-parent relations are established by linking each pixel at a given level (called *child*) to one of the pixels in the next higher level (called *parent*) which is closest in gray value (or in some other pixel attribute). The attribute values of the parents are then updated using the values of their children. The process of linking and updating is repeated until convergence (which always occurs [20]). At the end (or possibly during the linking process), some pixels are labeled as *roots*. In the simplest case, only pixels at the top level of the pyramid are roots. Every root and the pixels which are connected to it induce a tree in the pyramid. The leaves of each tree correspond to pixels in the full-resolution image which define a *segment* or *region*. Thus, the linked pyramid provides a framework for an iterative process of image segmentation. For example, in Fig. 7.2, pixel T is a root which represents a segment at the bottom level composed of pixels a, b, c and d.

There exist many variations on the scheme above. This may concern the way the initial pyramid is built, the manner pixels are linked to each other, the choice when pixels should be

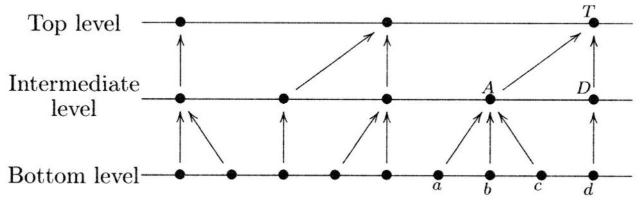


Figure 7.2: *A diagram illustrating linking relationships. E.g., pixel A is the parent of children a, b and c, and it is also the child of pixel T.*

declared as roots, the size of the neighborhood in which children can look for a parent to link to, the attribute that is being used (e.g., gray value, edge, local texture), etc.

As a result, a general pyramid linking method is hard to define, and most research has been focused on specific problems or aspects. Two major problems are the enforcement of connectivity in the segmented regions and the root labeling. The first problem arises from the fact that standard algorithms do not guarantee connectivity of regions. Pixels which are adjacent at some higher level do not necessarily represent adjacent regions at the lower levels. This can cause the creation of disconnected regions at the bottom level. To avoid such anomalies, one can use the *connectivity preservation criteria* proposed by Nacken in [113]. The second problem concerns root characterization. In Burt's original approach, only pixels at the top level are defined as roots, and therefore, the number of segments (which equals the number of roots) is fixed. Posterior approaches avoid such a prior choice and define roots as those pixels which are not linked 'strongly' enough to a parent [75, 164]. Now, the problem of root characterization is reduced to the definition of link strength and the choice of a root labeling threshold.

7.2.2 MR segmentation algorithm using linking

Our basic algorithm follows the classical '50% overlapping 4×4' structure [20]. This means that each parent is derived from the pixels in the 4×4 neighborhood immediately below it, and this neighborhood overlaps 50% of that of its 4 neighbors. Thus, each pixel has 16 candidate children and each child up to 4 candidate parents; see Fig. 7.3. The bottom of the pyramid corresponds to level zero and, for simplicity, is assumed to be of size $N \times N$ with N a power of 2. The highest level is considered to be $K_M = log_2 N - 1$.

At each level k, the pixels are indexed by the vector $\boldsymbol{n} = (m, n)^T$, where $m, n = 0, \ldots, \frac{N}{2^k} - 1$. We denote by $\mathcal{C}(\boldsymbol{n})$ the set of candidate children of pixel $\boldsymbol{n}$ at level $k > 0$; that is,

$$\mathcal{C}(\boldsymbol{n}) = \big\{(m', n') \mid m' \in \{2m-1, 2m, 2m+1, 2m+2\}, n' \in \{2n-1, 2n, 2n+1, 2n+2\}\big\}.$$

Similarly, we denote by $\mathcal{P}(\boldsymbol{n})$ the set of candidate parents of pixel $\boldsymbol{n}$ at level $k < K_M$:

$$\mathcal{P}(\boldsymbol{n}) = \big\{(m', n') \mid m' \in \{\lfloor\frac{1}{2}(m-1)\rfloor, \lfloor\frac{1}{2}m\rfloor, \lfloor\frac{1}{2}(m+1)\rfloor\}, n' \in \{\lfloor\frac{1}{2}(n-1)\rfloor, \lfloor\frac{1}{2}n\rfloor, \frac{1}{2}(n+1)\rfloor\}\big\},$$

where $\lfloor\cdot\rfloor$ denotes the integer part of the enclosed value. We define the *receptive field* of a pixel $\boldsymbol{n}$ as the set of pixels at level zero that belong to the same tree as $\boldsymbol{n}$.

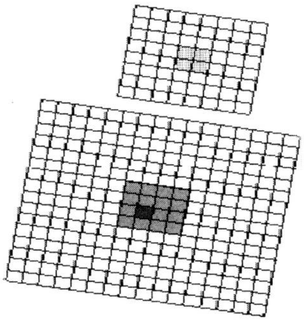


Figure 7.3: *Parent-child relations. The dark pixel at the lower level should choose a parent within the 4 candidate parents at the next finer level. Each of the candidate parents has 16 children. E.g., the bottom-left gray pixel at the finer level has as children the pixels shaded in gray in the lower level.*

To each pixel we associate one or more variables representing the attributes on which the segmentation will be based. In this study, we assign to each pixel $\boldsymbol{n}$ at level k its grayscale value $x^k(\boldsymbol{n})$, and the area $A^k(\boldsymbol{n})$ of its receptive field.

Consider an input image $x = x^0$. Our pyramid segmentation algorithm consists of three steps.

1. Initialization
We associate to each pixel $\boldsymbol{n}$ at level zero the gray value $x^0(\boldsymbol{n})$ of the original image, and to each pixel $\boldsymbol{n}$ at level $k > 0$ a gray value $x^k(\boldsymbol{n})$ computed from the average of the gray values of its candidate children:

$$x^k(\boldsymbol{n}) = \frac{1}{16} \sum_{\boldsymbol{n}' \in \mathcal{C}(\boldsymbol{n})} x^{k-1}(\boldsymbol{n}') .$$

2. Linking

(a) Pixel linking and root labeling.

For each child, a suitable parent is sought among the candidate parents: it is linked to its most 'similar' parent or it becomes a root (see below). Here, 'similarity' is based on grayscale proximity. A distance measure between the child and each of its four candidate parents is computed. A link is established with the parent that minimizes that distance. It may occur that more than one candidate parent minimizes the distance measure. In this case we arbitrarily pick one of them. A simple choice for the distance measure is the absolute difference in grayscale. Examples of other distances can be found in [113, 164].

In our approach, we perform the root labeling within the linking step. That is, when trying to link to a parent, the link is not established if the minimal distance is above some threshold. In such a case the pixel is labeled as a root (thus, it is not considered to be a child any more). We refer to [75, 164] for other alternatives.

An advantage of this method is its speed: a single operation will identify all roots. A disadvantage is that it is not clear beforehand how many roots (and therefore, how many segments) will be found. Defining a good root labeling threshold is not straightforward. When the threshold is too high, few pixels become roots, whereas many pixels are labeled as root if the threshold is too low. For simplicity, we use a threshold $T = 0.25\Delta x^0$ where Δx^0 is the length of the dynamic range of the input image x^0.

(b) Updating area A^k and gray values x^k.

The attributes of each parent are recomputed using only the children that are linked to it:

$$\begin{aligned} A^{k+1}(\boldsymbol{n}) &= \sum_{\boldsymbol{n}' \in \mathcal{C}(\boldsymbol{n})} A^k(\boldsymbol{n}') \\ x^{k+1}(\boldsymbol{n}) &= \frac{\sum_{\boldsymbol{n}' \in \mathcal{C}(\boldsymbol{n})} x^k(\boldsymbol{n}') A^k(\boldsymbol{n}')}{A^{k+1}(\boldsymbol{n})}, \end{aligned}$$

where $A^0(\boldsymbol{n}) = 1$ for all $\boldsymbol{n}$ at level zero.

(c) Iteration of (a) and (b) until convergence.

3. Segmentation

The actual segmentation is obtained by using the tree structures that have been created. At a given level k, pixels that are connected to a common root are classified as a single region segment. In this way, we obtain a segmentation $\mathcal{R}^k$ of the k'th-level approximation image.

7.2.3 MR/MS segmentation algorithm using linking

In the previous subsections we have discussed how to obtain an MR segmentation from a single input. Now we address the more difficult problem of how to compute a single MR segmentation based on multiple source images. We show that the segmentation method presented in the last subsection can be extended to the case where we have several input images x_S, $S \in \mathcal{S}$. In this case, the initialization step is performed as before for each image and, in the linking step, the distance between a child $\boldsymbol{n}$ and a candidate parent $\boldsymbol{n}' \in \mathcal{P}(\boldsymbol{n})$ is given by the expression

$$\left(\sum_{S \in \mathcal{S}} \left(x_S^k(\boldsymbol{n}) - x_S^{k+1}(\boldsymbol{n}') \right)^2 \right)^{1/2} . \tag{7.4}$$

As in the scalar case, the candidate $\boldsymbol{n}'$ which minimizes this distance is selected to become the parent unless the distance is above some threshold, in which case $\boldsymbol{n}$ is labeled as a root. Using the new links, the gray values are updated for each $S \in \mathcal{S}$, and the process of linking and updating is iterated until convergence. In this way, we obtain a single linked pyramid structure and we can apply the same segmentation step as before.

We summarize the basic steps of our MR/MS segmentation in the following algorithm.

Algorithm

1. For each input $S \in \mathcal{S}$
    - Construct an approximation pyramid $\{x_S^k\}$.
2. For each level $k < K_M$
    - While no convergence,
        * For each child $\boldsymbol{n}$ at level k, find parent $\boldsymbol{n}' \in \mathcal{P}(\boldsymbol{n})$ which minimizes the distance given by (7.4). If this distance is above some threshold, $\boldsymbol{n}$ is set as a root, otherwise it is linked to $\boldsymbol{n}'$.
        * For each parent $\boldsymbol{n}$ at level $k+1$, update $A^{k+1}(\boldsymbol{n})$ and $x_S^{k+1}(\boldsymbol{n})$ for all $S \in \mathcal{S}$.
3. For each level k
    - All pixels $\boldsymbol{n}$ at level k connected to a common root are classified as a single region segment in $\mathcal{R}^k$.

The segmentation is based on the approximation pyramids (computed from the grayscale values of the pixels) of the different input sources x_S, which are all treated equally. Obviously this is a very naive approach since different sources may present different amplitude ranges and may not be equally reliable. Thus, prior to segmentation, one may pre-process (e.g., normalization of amplitudes, denoising, etc.) the input images so that their attributes become comparable. Alternatively, one can modify the distance measure in (7.4) and use, for instance,

$$\left(\sum_{S \in \mathcal{S}} \mu_S \left(x_S^k(\boldsymbol{n}) - x_S^{k+1}(\boldsymbol{n}') \right)^2 \right)^{1/2} ,$$

where μ_S is a nonnegative normalization factor which may depend on several factors such as the dynamic range, noise estimation, entropy, etc.

Additionally, the segmentation algorithm can be improved by the use of connectivity preservation criteria, other root criteria, adaptive windows and probabilistic linking [113, 141, 164].

Note that, by construction, the MR segmentation $\mathcal{R}$ obtained with our algorithm has a pyramidal structure where the bottom level is at full resolution (same size as x_S) and each successive coarser level is 1/4 of its predecessor. However, this might not be true for the MR decompositions y_S obtained with the MR analysis block. Note also that the levels from the above MR segmentation $\mathcal{R}$ range from 0 to K_M, whereas the levels from the MR decompositions y_S go from 1 to K. In practice, K is smaller than K_M, so we assume henceforth that $K \leq K_M$. In addition, we assume that the MR/MS segmentation module associates to each image $y^k(\cdot|p)$, the partition $\mathcal{R}^{k'}$ such that they have the same dimensions and sampling structure. For instance, if y_S corresponds to a Laplacian decomposition, then $k' = k - 1$, for $k = 1, \ldots, K$; while if y_S corresponds to a discrete wavelet transform, then $k' = k$ for $k = 1, \ldots, K$ and all $p = 1, \ldots, 3$.

7.3 Experimental results

In this section, we present some experimental results obtained with one of the simplest implementations of the region-based fusion approach.

7.3.1 Case studies

We consider two input sources x_A and x_B. For their MR decomposition, we use a Laplacian pyramid (thus, we only have a single orientation band, i.e, $P = 1$). We employ the MR/MS segmentation algorithm discussed in Section 7.2. In the combination algorithm, we do not use a matching measure and define the activity of each region $R \in \mathcal{R}^k$ as in (7.1), with $a_S^k(\boldsymbol{n}|p) = |y_S^k(\boldsymbol{n}|p)|$. The combination process is performed as in (6.3), with $w_A(\delta) = \delta$ and $w_B(\delta) = 1 - \delta$. In the decision process, each component of d is obtained by the following simple decision rules:

- For $p = 0$,

$$\delta = d^K(\boldsymbol{n}|0) = \frac{1}{2}, \text{ for all } \boldsymbol{n} .$$

- For $p = 1$,
  - for each level k and for each region $R \in \mathcal{R}^k$:

$$\delta = d^k(\boldsymbol{n}|1) = \begin{cases} 1 & \text{if } a_A^k(R|1) > a_B^k(R|1) \\ 0 & \text{otherwise} \end{cases} \qquad \text{for all } \boldsymbol{n} \in R.$$

Note that according to this algorithm, the composite approximation image $y_F^K(\cdot|0)$ is the pixel-wise average of the approximation images $y_S^K(\cdot|0)$ and therefore, the region information $\mathcal{R}^K$ is neglected. The composite detail images y_F^k, however, are constructed by a selective combination as in (7.3).

We have tested our algorithm on several pairs of images. Three examples are given here to illustrate the fusion process described above. In all cases, we have chosen $K = 3$ and, when displaying the images, the gray values of the pixels have been scaled between 0 and 255 (histogram stretching). The first row of each figure shows the input sources x_A and x_B. The second row depicts the segmentation and decision map at the first level of decomposition. For the decision maps, black and white pixels correspond to $\delta = 0$ and $\delta = 1$, respectively. Thus, according to our algorithm, coefficients corresponding to 'white zones' are selected from y_A^k, while coefficients corresponding to 'black zones' are selected from y_B^k.

Case 1: fusion of visible and IR wavelength images - Fig. 7.4
Fig. 7.4 shows the input images, their corresponding segmentation and decision maps at levels 1 and 2, and the resulting composite image. It is interesting to note that, according to d^2 (bottom-middle of Fig. 7.4), although most of the background is selected from the visual image y_A^2, the region corresponding to the person is selected from the IR image y_B^2. Note that in the

composite image (bottom-right of Fig. 7.4) there is less contrast between the person and the background than in the IR image. This is due to the fact that the approximation images at the coarsest level are averaged, i.e., no region information has been used there.

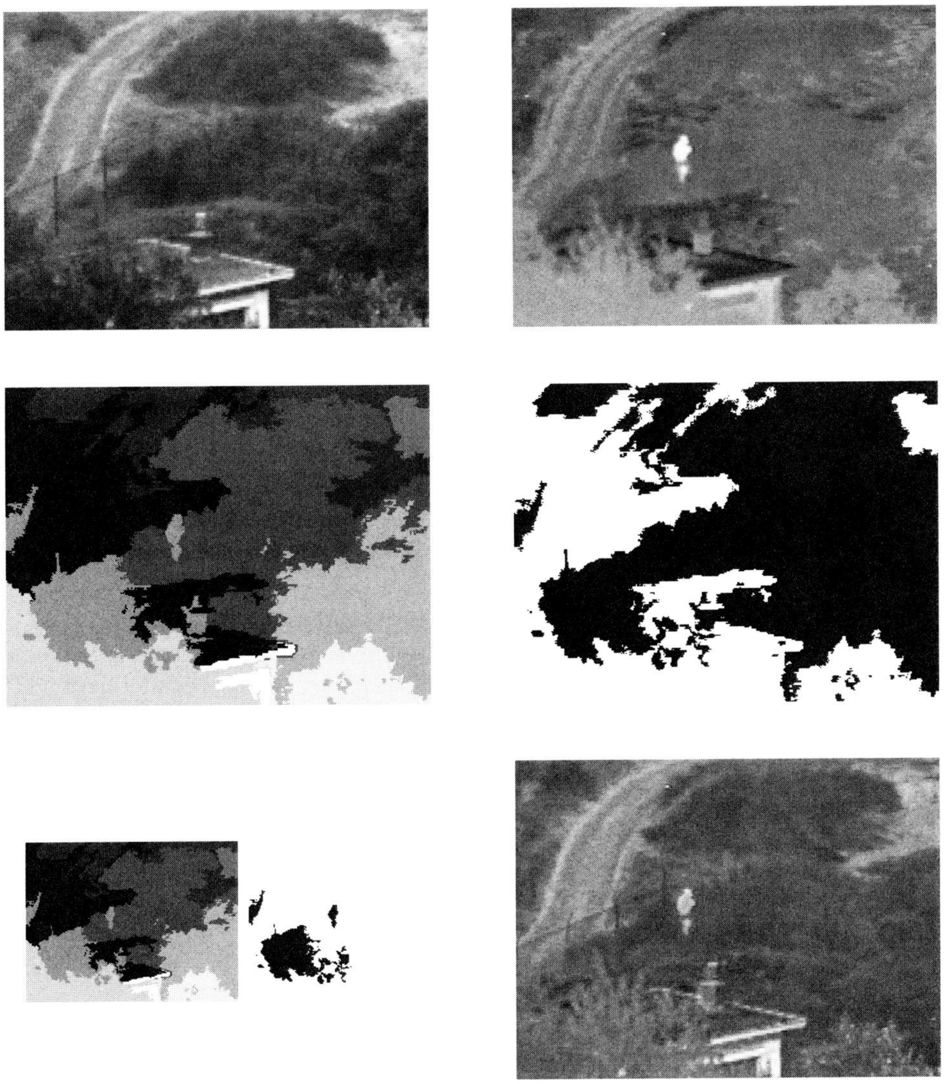


Figure 7.4: *Case* 1. *Top: visual (left) and IR (right) input images. Middle: segmentation (left) and decision map (right) at level* 1; *Bottom: segmentation (left) and decision map (middle) at level* 2, *and composite image (right).*

Case 2: fusion of images with different focus points - Fig. 7.5
As before, the second row (in Fig. 7.5) shows the segmentation and decision map at level 1. Note that since the digit '8' is connected to a particular region located within the left clock, the binary decision map d^1 points out, wrongly, to take the '8' from y_A^1 instead from y_B^1. The same happens at level $k = 2$ (not displayed here). The third row shows, from right to left, the resulting composite image and the composite image corresponding to a pixel-based MR fusion with the same fusion rules as in the region-based case. The bottom row illustrates how we can improve the region-based composite image by filtering the decision map. Here, we have filtered both decision maps d^1, d^2 with a morphological alternating filter: an opening followed by a closing [140]. The filtered d^1 is shown at the bottom left of Fig. 7.5. One can see that small details have been removed and that the boundaries have been smoothed. The composite image obtained with the filtered decision maps is shown at the bottom right.

Case 3: fusion of a magnetic resonance image (MRI) and a computer tomography (CT) image - Fig. 7.6
In this last example, we illustrate the combination of the approximation coefficients using an activity based on a local variance (see below). More precisely, we perform the selective combination in (7.3) for both detail and approximation coefficients but using different activity measures. For the details, $a_S^k(R|1)$ is defined as before, while for the approximation we consider

$$a_S^K(R|0) = \frac{1}{|R|} \sum_{\boldsymbol{n} \in R} \left(y_S^K(\boldsymbol{n}|0) - \bar{y}_S^K(R|0) \right)^2 ,$$

where $\bar{y}_S^K(R|0) = \frac{1}{|R|} \sum_{\boldsymbol{n} \in R} y_S^K(\boldsymbol{n}|0)$.

Fig. 7.6 shows the input images, their corresponding segmentation and decision maps at levels 1, 2 and 3, the resulting composite image and, for comparison, two other composite images obtained with different algorithms than the above. By visual inspection, we can see that the proposed region-based fusion (right image of the third row) preserves the soft tissue depicted in the MRI image better than the pixel-based fusion (left image of the bottom row). The right image of the bottom row is the composite image resulting from a fusion algorithm where the region-based approach is only used for the detail images (as we did in the previous experiments).

7.3.2 Discussion

From the experiments presented, we can see that, despite the crudeness of the current implementation, the visual performance is surprisingly good. This suggests that the region-based approach proposed here can at least be competitive with (but more likely outperform) other MR fusion techniques.

The importance of the various parameters in our approach is a topic that needs much more investigation. In particular, our MR/MS segmentation is very sensitive to the root labeling criteria and the threshold proposed in Section 7.2.2 does not always give a satisfactory segmentation.

Further investigations are necessary for the fine-tuning of parameters as well as for the proper selection of the different ingredients of the scheme. Toward this end, performance assessment criteria have been developed (see Chapter 8) to evaluate and demonstrate the capacities of the new fusion technique, as well as to compare its performance with other MR fusion schemes.

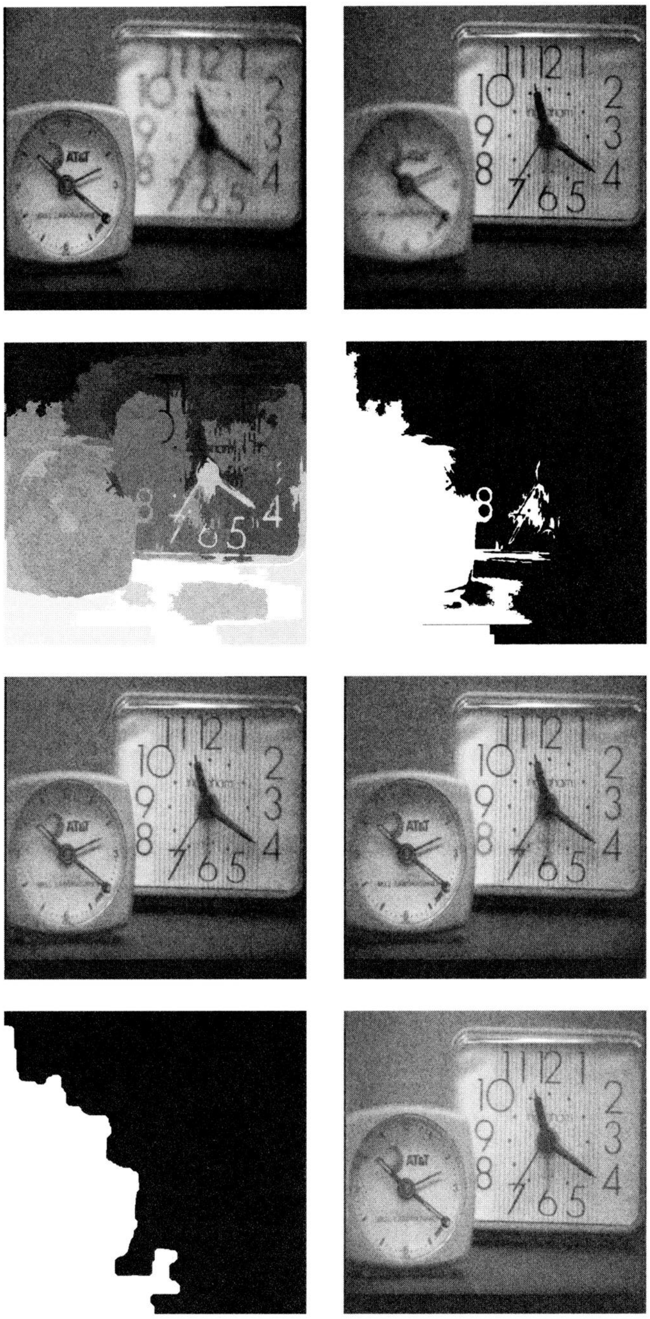


Figure 7.5: *Case 2. Top: multi-focus input images. Second row: segmentation (left) and decision map (right) at level* 1. *Third row: composite images with pixel-based (left) and region-based (right) approach. Bottom: filtered decision map (left) and corresponding composite image (right).*

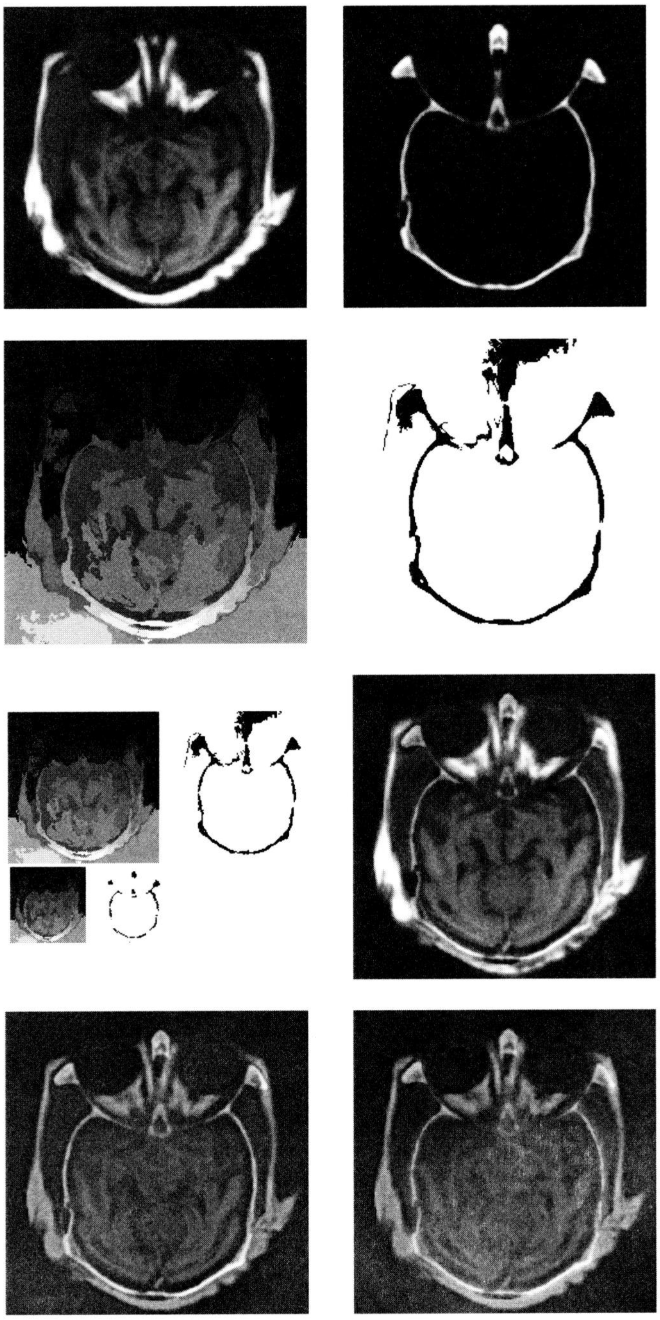


Figure 7.6: *Case* 3. *Top: MRI (left) and CT (right) input images. Second row: segmentation (left) and decision map (right) at level* 1. *Third row: segmentation (left) and decision map (middle) at levels* 2 *and* 3, *and composite image (right). Bottom: pixel-based (left) and region-based (right) composite images with pixel-wise average combination for the approximation images.*

Chapter 8

Performance assessment in image fusion

The widespread use of image fusion methods, in military applications, in surveillance, in medical diagnostics, etc., has led to an increasing need for pertinent performance or quality assessment tools in order to compare results obtained with different algorithms or to obtain an optimal setting of parameters for a given fusion algorithm.

In most cases, image fusion is only a preparatory step to some specific task such as human monitoring, and thus the performance of the fusion algorithm has to be measured in terms of improvement of the subsequent tasks. For example, in classification tasks, a common evaluation measure is the percentage of correct classifications. This requires that the 'true' correct classifications are known. In experimental setups, however, the availability of a ground-truth is not guaranteed.

In this chapter, we focus on general performance measures which can be computed independently of the subsequent task. More precisely, we are interested in measures that express the successfulness of an image fusion technique by the extent that it creates a composite that retains salient information from the sources while minimizing the number of artifacts or the amount of distortion that could interfere with interpretation.

In Section 8.1, we present a brief summary of the state-of-the art methods for measuring fusion performance. In Section 8.2, we propose three variants of a new quality measure for image fusion. The interest of our measures, which are based on an image quality index recently introduced by Wang and Bovik in [166], lies in the fact that they do not require a ground-truth or reference image. In Section 8.3, we perform several simulations which show that our measures are compliant with subjective evaluations and can therefore be used to compare different image fusion methods or to find the best parameters for a specific fusion algorithm. We also use these measures to demonstrate the effectiveness of the region-based fusion approach described in Chapter 7.

8.1 Existing approaches to image fusion performance

In many applications, the end user or interpreter of the fusion result is a human. Thus, the human perception of the composite image is of paramount importance and therefore, fusion results are mostly evaluated by subjective criteria [124, 156]. This involves human observers to judge the *quality* of the resulting composite images. As such a 'human quality measure' depends highly on psychovisual factors, these subjective tests are difficult to reproduce or verify. Furthermore, they are time consuming and expensive. This shows clearly the need for objective measures that quantify the performance of fusion algorithms. The key problem is how to quantify a subjective impression like image quality. One way to 'solve' this problem is by associating quality with the deviation of the experimental composite image from the 'ideal' composite image [92, 165, 173]. Then, another problem arises, namely, how to define the 'ideal' composite image. A less common approach is to design performance measures which, without assuming knowledge of a ground-truth, can be used for quality assessment of the composite image by quantifying the degree to which the composite image is 'related' to the input sources [117, 172].

Examples of reference-based quality measures for fusion

The work by Li *et al.* [92] is an example where out-of-focus image fusion is evaluated by comparison of the composite image with an 'ideal' composite created by a manual 'cut and paste' process. Indeed, various fusion algorithms presented in the literature have been evaluated by constructing some kind of ideal composite image x_R and comparing with the experimental composite result x_F. In this case, fusion evaluation amounts to measure the distortion or dissimilarity between x_R and x_F. Generally speaking, the smaller the distortion, the better the quality of the composite image.

The l^2-metric, also called *root mean squared error*[1], given by

$$d_2(x_R, x_F) = \left(\frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \left(x_R(m,n) - x_F(m,n) \right)^2 \right)^{1/2} \tag{8.1}$$

(where M, N are the dimensions of the images), is widely used for such purposes, notwithstanding its well-known limitations. In a certain way, it measures the total amount of energy distortion. High errors correspond to high distortions. If the images take values between $x_{\min}$ and $x_{\max}$, then d_2 has a dynamic range of $[0, x_{\max} - x_{\min}]$.

Another class of measures is based on concepts from information theory [37]. The empirical *mutual information* is often used for fusion evaluation [165, 173]:

$$I(x_R; x_F) = \sum_{u=1}^{L} \sum_{v=1}^{L} p_{R,F}(u,v) \log_2 \frac{p_{R,F}(u,v)}{p_R(u) p_F(v)}, \tag{8.2}$$

where p_R, p_F are the normalized graylevel histograms of x_R, x_F, respectively, $p_{R,F}$ is the joint graylevel histogram of x_R and x_F, and L is the number of bins; see Section 5.4. The measure $I(x_R; x_F)$ indicates how much information the composite image x_F conveys about the reference x_R. Thus, the higher the mutual information between x_F and x_R, the more x_F resembles the ideal x_R. In this sense, mutual information can be interpreted as a 'similarity' measure,

---

[1] In Chapter 5, we introduced the mean squared error (MSE) which is just the squared l^2-metric; see formula (5.1).

in contrast with the l^2-metric in (8.1) which can be seen as a 'dissimilarity' (i.e., distortion) measure. Note that $I(x_R; x_F)$ has a dynamic range of $[0, \min\{H(x_R), H(x_F)\}]$, where H is the empirical entropy (see Section 5.4). Ideally, one has $I(x_R; x_F) = H(x_R)$, although this does not imply that $x_R = x_F$.

Examples of non-reference quality measures for fusion

An example of an objective performance measure which does not assume the knowledge of a ground-truth is given by Xydeas and Petrović in [172]. Their performance measure models the accuracy with which visual information is transferred from the source images to the composite image. In their approach, important visual information is associated with edge information measured for each pixel. Thus, they measure the fusion performance by evaluating the relative amount of edge information that is transferred from the input images to the composite image. This amount is normalized to the range $[0, 1]$, so that value 0 corresponds to the 'complete loss' of edge information from the sources to the composite x_F, and value 1 to the 'total preservation'.

Another non-reference objective performance measure is proposed by Qu *et al.* in [117]. They evaluate fusion performance by adding the mutual information between the composite image and each of the input images, i.e., they compute

$$I(x_A; x_F) + I(x_B; x_F)\,, \tag{8.3}$$

where $I(x_S; x_F)$ is computed as in (8.2). The higher the value in (8.3), the better the quality of the composite image is supposed to be.

Each of the aforementioned approaches has its pro's and contra's, but it is fair to conclude that objective performance assessment in fusion is largely an open problem which has received relatively little attention. Most existing performance assessment methods are low-level, i.e., they act on the pixel level. High-level methods, i.e., acting on region or even object level are non-existent to the best of our knowledge. More research is required to provide valuable objective evaluation methods for image fusion, in particular, where it concerns region or object-based methods.

8.2 A new quality measure for image fusion

This section discusses a novel objective non-reference quality assessment method for image fusion that utilizes local measurements to estimate how well salient information contained within the sources is represented by the composite image. Our quality measure is based on an image quality index proposed by Wang and Bovik in [166].

8.2.1 The image quality index of Wang and Bovik

We present a brief introduction to the image quality index that was recently introduced by Wang and Bovik in [166]. Given two images x and y of size $M \times N$, let $\bar{x}$ denote the mean of

x, let σ_x^2 and σ_{xy} be the variance of x and covariance of x, y, respectively, i.e.,

$$\begin{aligned} \sigma_x^2 &= \frac{1}{MN-1} \sum_{m=1}^{M} \sum_{n=1}^{N} \big(x(m,n) - \bar{x}\big)^2 \\ \sigma_{xy} &= \frac{1}{MN-1} \sum_{m=1}^{M} \sum_{n=1}^{N} \big(x(m,n) - \bar{x}\big)\big(y(m,n) - \bar{y}\big) . \end{aligned}$$

Define

$$Q_0 = \frac{4\sigma_{xy}\bar{x}\bar{y}}{(\bar{x}^2 + \bar{y}^2)(\sigma_x^2 + \sigma_y^2)} ,$$

which can be decomposed as

$$Q_0 = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \cdot \frac{2\bar{x}\bar{y}}{\bar{x}^2 + \bar{y}^2} \cdot \frac{2\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2} . \tag{8.4}$$

Wang and Bovik refer to Q_0 as an *image quality index* and use it to quantify the structural distortion between images x and y, one of them being the reference image and the other the distorted one. In fact, the value $Q_0 = Q_0(x, y)$ is a measure for the similarity of images x and y and takes values between -1 and 1. Note that the first component in (8.4) is the correlation coefficient between x and y. The second component corresponds to a kind of average luminance distortion and it has a dynamic range of $[0, 1]$ (assuming nonnegative mean values). The third factor in (8.4) measures a contrast distortion and its range is also $[0, 1]$. The maximum value $Q_0 = 1$ is achieved when x and y are identical.

Since image signals are generally non-stationary, it is appropriate to measure the number Q_0 over local regions and then combine the different results into a single measure. In [166] the authors propose to use a sliding window approach: starting from the top-left corner of the two images x, y, a sliding window of fixed size moves pixel by pixel over the entire image until the bottom-right corner is reached. For each window w, the local quality index $Q_0(x, y \mid w)$ is computed for the values $x(m, n)$ and $y(m, n)$ where pixels (m, n) lie in the sliding window w. Finally, the overall image quality index Q_0 is computed by averaging all local quality indices:

$$Q_0(x, y) = \frac{1}{|W|} \sum_{w \in W} Q_0(x, y \mid w) , \tag{8.5}$$

where W is the family of all windows and $|W|$ is the cardinality of W.

Wang and Bovik [166] have compared (under several types of distortions) their quality index with existing image measures such as the mean squared error (MSE) as well as with subjective evaluations. Their main conclusion was that their new index outperforms the MSE, and they believe this to be due to the index's ability of measuring structural distortions, in contrast to the MSE which is highly sensitive to the l^2-energy of errors.

8.2.2 A new fusion quality measure

We use the Wang-Bovik image quality index Q_0 defined in (8.5) to define a quality measure $Q(x_A, x_B, x_F)$ for image fusion. Here x_A, x_B are two input images and x_F is the composite image. The measure $Q(x_A, x_B, x_F)$ should express the 'quality' of the composite image given the inputs x_A, x_B.

We denote by $s(x_A|w)$ the saliency of image x_A in window w. It should reflect the local relevance of image x_A within the window w, and it may depend on, e.g., contrast, variance, or entropy. Given the local saliencies $s(x_A|w)$ and $s(x_B|w)$ of the two input images x_A and x_B, we compute a local weight $\lambda(w)$ between 0 and 1 indicating the relative importance of image x_A compared to image x_B: the larger $\lambda(w)$, the more weight is given to image x_A. A typical choice for $\lambda(w)$ is

$$\lambda(w) = \frac{s(x_A|w)}{s(x_A|w) + s(x_B|w)}. \tag{8.6}$$

Now we define the fusion quality measure $Q(x_A, x_B, x_F)$ as

$$Q(x_A, x_B, x_F) = \frac{1}{|W|} \sum_{w \in W} \Big(\lambda(w) Q_0(x_A, x_F|w) + \big(1 - \lambda(w)\big) Q_0(x_B, x_F|w) \Big). \tag{8.7}$$

Thus, in regions where image x_A has a large saliency compared to x_B, the quality measure $Q(x_A, x_B, x_F)$ is mainly determined by the 'similarity' of x_F and input image x_A. On the other hand, in regions where the saliency of x_B is much larger than that of x_A, the measure $Q(x_A, x_B, x_F)$ is determined mostly by the 'similarity' of x_F and input image x_B.

At this point, our model has produced a quality measure which gives an indication of how much of the salient information contained in each of the input images has been transferred into the composite image. However, the different quality measures obtained within each window have been treated equally. This is in contrast with the human visual system which is known to give higher importance to visually salient regions in an image. We now define another variant of the fusion quality measure by giving more weight to those windows where the saliency of the input images is higher. These correspond to areas which are likely to be perceptually important parts of the underlying scene. Therefore the quality of the composite image in those areas is of more importance when determining the overall quality. The overall saliency of a window is defined as $C(w) = \max\{s(x_A|w), s(x_B|w)\}$. The *weighted fusion quality measure* is then obtained as

$$Q_W(x_A, x_B, x_F) = \sum_{w \in W} c(w) \Big(\lambda(w) Q_0(x_A, x_F|w) + \big(1 - \lambda(w)\big) Q_0(x_B, x_F|w) \Big), \tag{8.8}$$

where $c(w) = C(w) / \big(\sum_{w' \in W} C(w') \big)$. There are various other ways to compute the weights $c(w)$ (for example, we could define $C(w) = s(x_A|w) + s(x_B|w)$), but we have found that the choice made here is a good indicator of important areas in the input images.

We introduce one final modification of the fusion quality measure that takes into account some aspect of the human visual system, namely the importance of edge information. Note that we can evaluate Q_W in (8.8) using 'edge images' (e.g., the norm of the gradient) instead of the original grayscale images x_A, x_B and x_F. Let us denote the edge image corresponding with x_A by x'_A. Now we combine $Q_W(x_A, x_B, x_F)$ and $Q_W(x'_A, x'_B, x'_F)$ into a so-called *edge-dependent fusion quality measure* by

$$Q_E(x_A, x_B, x_F) = Q_W(x_A, x_B, x_F)^{1-\alpha} \cdot Q_W(x'_A, x'_B, x'_F)^{\alpha}, \tag{8.9}$$

where the parameter $\alpha \in [0, 1]$ expresses the contribution of the edge image compared to the original image: the closer α is to one, the more important is the edge image.

Note that the three proposed measures have a dynamic range of $[-1, 1]$. The closer the value to 1, the higher the quality of the composite image.

8.3 Experimental results

In this section we use the proposed fusion quality measures defined in (8.7), (8.8) and (8.9) to evaluate different multiresolution (MR) image fusion schemes (see Chapters 6-7).

In the computation of the quality measures defined in last section, we take $\lambda(w)$ as in (8.6), with $s(x_A|w)$, $s(x_B|w)$ being the variance of images x_A and x_B, respectively, within the window w of size 8×8. In all displayed images, we have performed a histogram stretching and we have scaled the gray values of the pixels between 0 (black) and 255 (white).

8.3.1 Case studies

In the next two experiments, we present some results using the Laplacian pyramid, the ratio pyramid and the spatially-invariant discrete wavelet transform (SIDWT) as MR transforms of the input sources. In all cases we perform a 3-level decomposition. We combine the coefficients of the MR decompositions of each input by selecting at each position the coefficient with a maximum absolute value, except for the approximation coefficients from the lowest resolution where we take the average. For comparison, we also use the simple fusion method of averaging the input images.

Case 1: fusion of complementary blurred images - Fig. 8.1
First, we take as input images the complementary pair shown in the top row of Fig. 8.1. They have been created by blurring the original 'Cameraman' image of size 256×256 with a disk of diameter of 11 pixels. The images are complementary in the sense that the blurring occurs at the left half and the right half, respectively. In the second row we display their total weights used to compute Q_W in (8.8). More specifically, each pixel (m, n) in the left image contains the value $c(w)\lambda(w)$ with w being the window whose top-left corner corresponds to (m, n). Similarly, the right image displays $c(w)\big(1 - \lambda(w)\big)$ for every $w \in W$. The composite images obtained by the Laplacian pyramid, the ratio pyramid, the SIDWT and the average are depicted in the third and fourth row, from left to right. Table 8.1 compares the quality of these composite images using our proposed quality measures. The first row corresponds to the fusion quality measure Q defined in (8.7), the second row to the weighted fusion quality measure Q_W in (8.8) and the third row to the edge-dependent fusion quality measure Q_E in (8.9) with $\alpha = 1/2$. For

comparison, we also compute the l^2-metric in (8.1) between the original 'Cameraman' image and each of the composite images. Note that in 'real' fusion scenarios we do not have access to the original image. The resulting errors are shown in the last row of Table 8.1.

Fig. 8.1 shows that the Laplacian and SIDWT methods are comparable and that they outperform the other two schemes. Note, for instance, the blurring (e.g., in the buildings) and the loss of texture (e.g., in the grass) of the composite images obtained by the ratio pyramid and averaging. Furthermore, in the ratio-pyramid composite image, the details of the man's face have been cleared out, and in the average composite image, the loss of contrast is evident. These subjective visual comparisons are corroborated by the results in Table 8.1. Note that the Laplacian method has a higher Q_E than the SIDWT. This is most likely due to the fact that the former method is better able to preserve edges and reduce the ringing artifacts around them.

measure	Laplacian	Ratio	SIDWT	Average
Q	0.903	0.764	**0.930**	0.830
Q_W	0.962	0.827	**0.965**	0.874
Q_E	**0.966**	0.781	0.962	0.689
d_2	**8.41**	164.35	13.03	30.66

Table 8.1: *Comparison between different quality measures for the composite images in Fig. 8.1.*

Case 2: fusion of a magnetic resonance image (MRI) and a computer tomography (CT) image - Fig. 8.2

Consider now the input images in the top row of Fig. 8.2. We repeat the same computations as described above. The results are shown in Fig. 8.2 and Table 8.2. In this case, however, as we do not have a reference image to compare with, we cannot compute the l^2-metric. Instead, we use a measure based on mutual information. More precisely, the results in the last row of Table 8.2 have been obtained by adding the mutual information between the composite image and each of the inputs, such as in (8.3), and dividing it by the sum of the entropies of the inputs, i.e.,

$$\mathrm{MI}(x_A, x_B, x_F) = \frac{I(x_A; x_F) + I(x_B; x_F)}{H(x_A) + H(x_B)}.$$

In this way, we normalize the measure in (8.3) to the range $[0, 1]$.

In Fig. 8.2, we can see that again the Laplacian and SIDWT methods clearly outperform the other two methods. For both of them, many details (specially the brain tissue in the magnetic resonance image) have been lost. Moreover, due to the high contrast in the input images, the ratio pyramid blows up the dynamic range for some pixels, which makes it necessary to clip them in order to be able to 'visualize' the image. Again, the subjective visual analysis is consistent with the new quality indices, as shown in Table 8.2. In both experiments, the edge-dependent fusion quality measure gives a stronger separation between the good results (Laplacian and SIDWT) and the bad results (ratio and average). Note that the last row, where mutual information has been used, gives the best ranking to the average fusion method. However, mutual information has been shown to be a good indicator of the quality of MR composite images [117] (as long as the average is not taken in all levels for the construction of the composite MR decomposition).

measure	Laplacian	Ratio	SIDWT	Average
Q	0.661	0.601	**0.699**	0.636
Q_W	**0.799**	0.673	0.770	0.642
Q_E	**0.834**	0.645	0.814	0.608
MI	0.337	0.221	0.409	**0.691**

Table 8.2: *Comparison between different quality measures for the composite images in Fig. 8.2.*

Case 3: region-based vs. pixel-based MR fusion approach - Fig. 8.3
Finally, we use our quality measures to evaluate the region-based composite images obtained in Section 7.3 and which we redisplay in the left column of Fig. 8.3. We recall that in the 'Surveillance' and 'Clock' cases (Case 1 and Case 2 of Section 7.3, respectively), the region information was only used for the combination of the detail images, while for the 'Skull' case (Case 3 of Section 7.3) the region information was also used for the combination of the approximation images. Note also that in the 'Clock' case, we take the region-based composite image obtained with the post-processed decision map (see images at the bottom of Fig. 7.5). To get an impression of the potential of the region-based fusion approach, we also compute the quality measures for the composite image resulting from a pixel-based fusion (see right column of Fig. 8.3). The results are shown in Table 8.3. One can see that for the 'Skull' case the region-based approach yields higher quality values than the pixel-based approach. For the other two cases, the quality values obtained for the region-based scheme are comparable to the ones obtained for the pixel-based scheme and, in some cases, slightly lower. This is probably because, for the 'Surveillance' and 'Clock' cases, the composite approximation image was obtained by just averaging the approximation images of the inputs, thus disregarding the region information.

Our conclusion therefore is that in most cases the region-based scheme outperforms the pixel-based scheme. Moreover, we can infer that the use of region information for the combination of the approximation images as well as for the detail images (as used in the 'Skull' case) improves the composite image substantially.

8.3.2 Discussion

In this chapter we have discussed some new objective quality measures for image fusion which do not require a reference image and correlate well with subjective criteria as well as with other existing performance measures. Our measures are easy to calculate and applicable to various input modalities (and hence to different fusion applications). In particular, our measures give good results on variable quality input images since they take into account the locations as well as the magnitude of the distortions.

Further research is necessary to study the influence of the different parameters of the measures (e.g., size of the window, choice of saliency and weights, etc.), and how to select them in

measure & method		Surveillance	Clock	Skull
Q	region-based	**0.645**	0.954	**0.854**
	pixel-based	0.632	**0.955**	0.694
Q_W	region-based	0.645	**0.969**	**0.812**
	pixel-based	**0.646**	0.961	0.746
Q_E	region-based	**0.444**	0.834	**0.619**
	pixel-based	0.427	**0.841**	0.608
MI	region-based	**0.108**	**0.497**	**0.551**
	pixel-based	0.103	0.493	0.364

Table 8.3: *Comparison between region-based and pixel-based fusion for the composite images in Fig. 8.3.*

order to optimize the quality measures.

There are several areas in which our quality measures can be extended. We currently consider grayscale images, so inclusion of color is an obvious extension. Other visual mechanisms of our visual system may also be taken into account. One such mechanism is multiresolution. Since the sensitivity of the human visual system varies over spatial frequencies, it seems natural to compute the quality measures with respect to the scales of the objects that appear in the image. Another possible extension is motivated by our work on region-based fusion. Rather than calculating the quality measure in fixed windows, one might choose to segment the sources first and compute the measure region by region.

In addition, we plan to include some information-theoretic measures such as mutual information and entropy to better estimate the information content of the composite image. We also plan to study how our objective measures can be used to guide a fusion algorithm and improve the fusion performance.

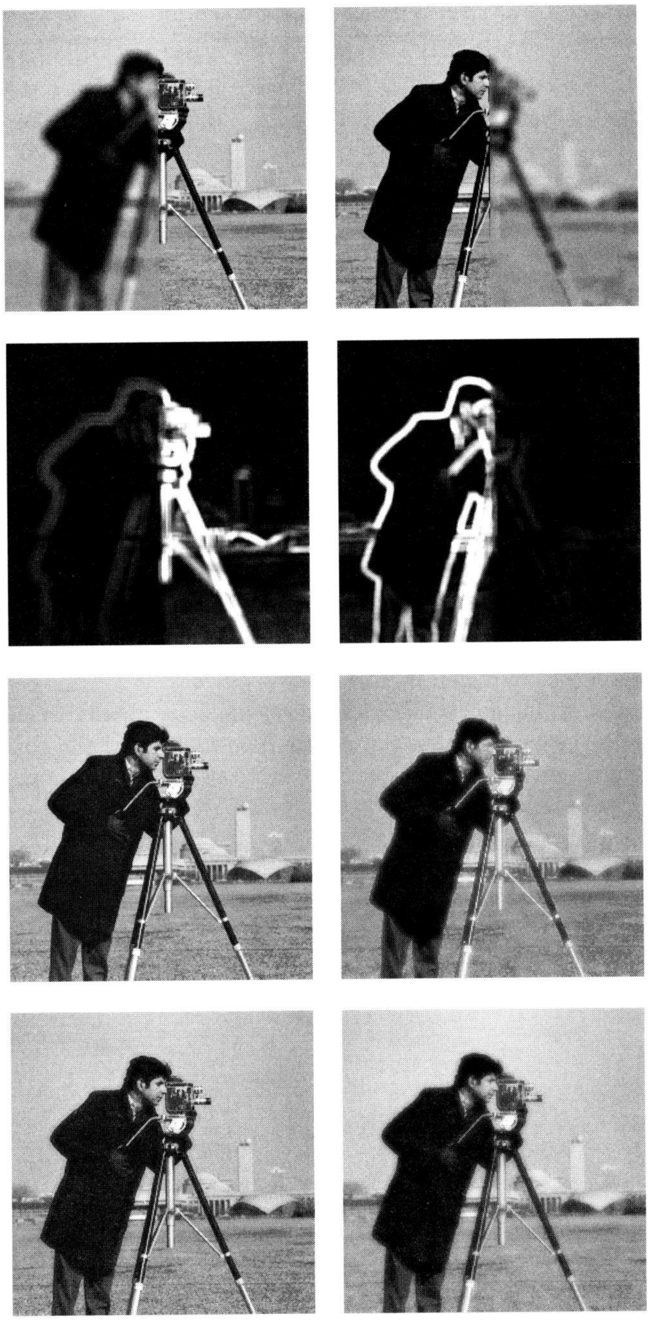


Figure 8.1: *Case* 1. *Top: input images x_A (left) and x_B (right). Second row: total weights $c \cdot \lambda$ (left) and $c \cdot (1-\lambda)$ (right). Third row: composite images with a Laplacian (left) and a ratio (right) pyramid decompositions. Bottom: composite images with a SIDWT (left) decomposition and averaging (right).*

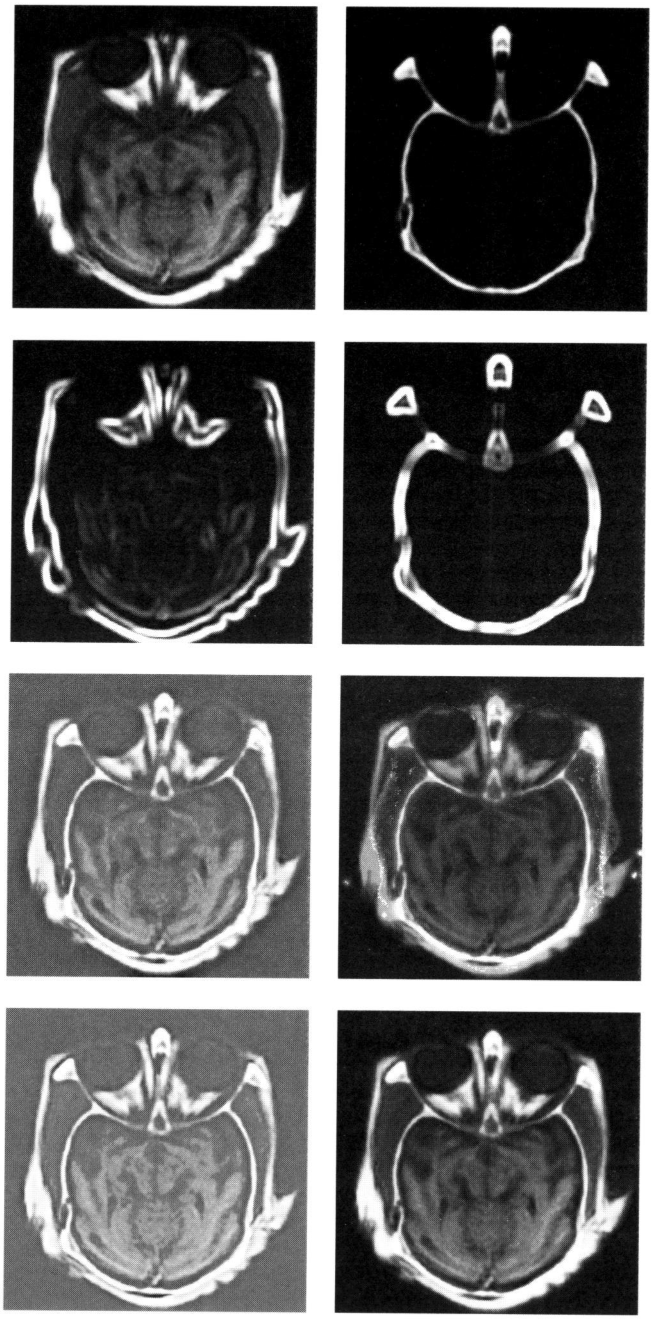


Figure 8.2: *Case 2. Top: input images x_A (MRI, left) and x_B (CT image, right). Second row: total weights $c \cdot \lambda$ (left) and $c \cdot (1 - \lambda)$ (right). Third row: composite images with a Laplacian (left) and a ratio (right) pyramid decompositions. Bottom: composite images with a SIDWT (left) decomposition and averaging (right).*

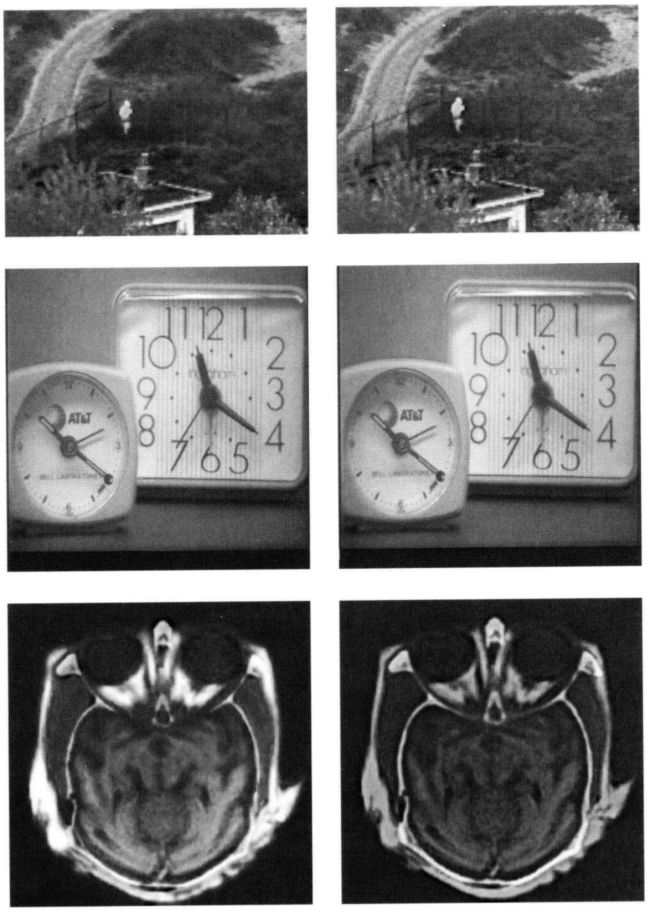


Figure 8.3: *Case* 3. *Region-based (left) and pixel-based (right) composite images. See also Fig. 7.4-Fig. 7.6.*

Chapter 9

Conclusions

In this thesis we have described a framework for multiresolution processing of signals and images. One of our main contributions is the development of non-redundant adaptive wavelets with perfect reconstruction. Another important contribution is the description of an axiomatic setup for multiresolution-based fusion. Within this formalism, we have developed a new region-based image fusion approach. Finally, we have introduced a quality measure for image fusion to assess the performance of the various fusion algorithms.

The results obtained in this thesis have been discussed at length in the various chapters. In the following, we summarize the main results and conclusions.

In Chapter 2, we have presented an axiomatic multiresolution decomposition scheme with perfect reconstruction. Many multiresolution signal decomposition schemes proposed in the literature are special cases of the general schemes discussed here. The lifting scheme, usually restricted to wavelet decompositions, has been extended. In particular, it has been shown how lifting can be used in the context of pyramid decompositions.

In Chapters 3-5, we have introduced an adaptive wavelet decomposition based on an adaptive update lifting step. The underlying idea is to choose the update lifting filters according to some decision criterion which depends on local characteristics of the signal. An important feature of our adaptive wavelet decomposition scheme is that it is neither causal nor requires any bookkeeping in order to perform perfect reconstruction.

The particular case where the system can choose between two different update filters based on the local gradient has been thoroughly discussed. In Sections 4.1-4.3, we have derived necessary and sufficient conditions for the invertibility of such adaptive schemes for various scenarios. Several simulation results have been given to show the differences between our adaptive decompositions and the non-adaptive decompositions based on a fixed lifting scheme. These examples illustrate the potential of adaptive schemes for preserving the discontinuities in signals and images even at low resolutions. Furthermore, it has been shown in Section 5.4 that adaptive schemes often yield decompositions that have lower entropies than schemes with fixed update filters, a property that is highly relevant in the context of compression. In Section 5.5, we have analyzed the quantization effects on our adaptive scheme. In fact, we have been able to derive conditions that guarantee perfect reconstruction of the decision map (i.e., the choice of the update filters).

Despite all these attractive properties, a number of open theoretical and practical questions need to be addressed before such schemes become useful in signal processing and analysis applications. For example, we need to get a better understanding how to design update and prediction operators that lead to adaptive wavelet decompositions that satisfy properties key to a given application at hand. In this thesis, we have focused on binary decision maps and as a result, the adaptive scheme can only discriminate between two 'geometric events' (e.g., edge region or homogeneous region). In order to deal with the great richness of real-world signals and images, one must be able to incorporate the geometrical structure of the signals, for example, by using multiple criteria.

Another issue that needs to be addressed is the stability of the scheme. In particular, the behavior of the adaptive scheme under quantization needs a more thorough investigation. Stability of decompositions is of utmost importance when they are being used in lossy compression schemes for image or video coding.

In order to evaluate the potential of our adaptive schemes in applications such as compression, it will be necessary to make a systematic comparison with existing decompositions schemes (such as the ones described in Section 3.1).

Another future research effort will be the extension of the current framework to adaptive prediction lifting steps. Moreover, it would be worthwhile to see if our adaptive scheme can be extended to include also morphological filters.

Chapters 6-8 have been devoted to image fusion. We have introduced an axiomatic framework for multiresolution image fusion. The proposed framework not only encompasses most of the existing multiresolution image fusion schemes, but also allows the construction of new ones, both pixel and region-based.

The region-based fusion scheme presented in Chapter 7 is an extension of the classical pixel-based schemes. The basic idea is to perform a multiresolution/multisource segmentation of the various input images in order to guide the fusion process. For this purpose, we developed a multiresolution/multisource segmentation method based on pyramid linking and suggested some combination algorithms which make use of the resulting segmentation. Several experimental results have been shown.

The implementation of our region-based fusion approach is still in a preliminary stage and in the experiments performed we did not attempt to optimize its performance. However, the results obtained so far suggest that our approach may be useful for several image fusion applications. We need to investigate this more thoroughly in the future. In particular, we plan to study the effect of the different parameters and functions in the scheme on the final fusion process. We also intend to design new combination algorithms and replace the current segmentation by pyramid linking by some other techniques, such as hierarchical watersheds from mathematical morphology.

A substantial part of our efforts will be devoted to the optimization and extension of the objective measures for image fusion described in Chapter 8. We intend to use such objective measures to evaluate and demonstrate the capacities of our region-based fusion approach, as well as to compare its performance with other multiresolution fusion schemes. We also plan to explore how these objective measures can be used to guide fusion and improve fusion performance.

Finally, another interesting research area concerns the development of adaptive fusion techniques in which the source images determine the type of multiresolution transform being used as well as the parameters that are involved.

Appendix A

MATIFUS: a MAtlab Toolbox for Image FUSion

MATIFUS[1] is a Matlab-based toolbox for image fusion with a graphical user interface. It contains numerous routines for multiresolution-based image fusion, and provides a user friendly interface to the image fusion framework proposed in Chapters 6 and 7. Thus, it allows both pixel-based and region-based fusion approaches. MATIFUS is easily extendible in the sense that new types of multiresolution decompositions and fusion rules can be added. Furthermore, MATIFUS facilitates the setting of fusion methods and parameters, and allows to study the influence of such settings on the visual quality of the fusion results, enabling the users to choose the method which is the most appropriate for their particular application.

In the following, we give a very brief overview of the MATIFUS toolbox. The corresponding documentation and website[2] are in preparation.

Functionalities

MATIFUS consists of two main panels: the *control panel*, which contains the settings and functions related to the fusion process, and the *visualization panel*, which displays the inputs images, the fused image and other intermediate results such as the segmentation and decision maps.

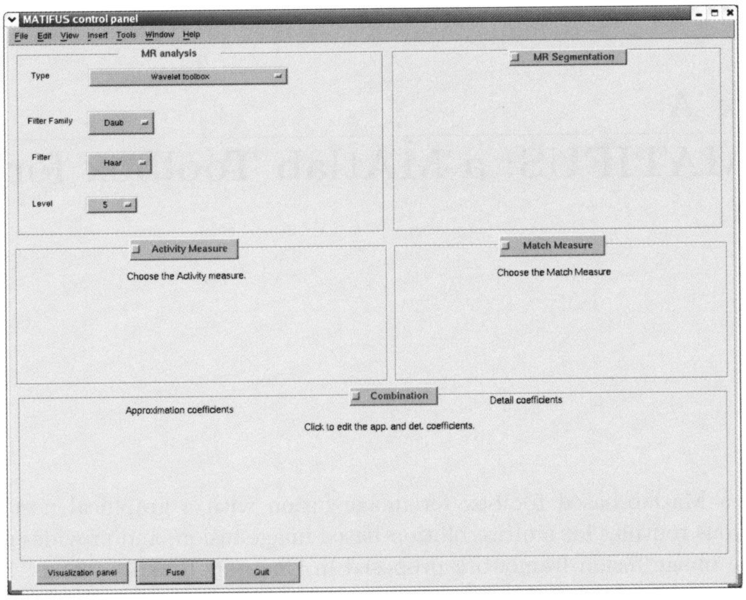
Fig. A.1 shows the control panel with the default setting parameters. The user can construct several fusion algorithms by the interactive selection of the different parameters in the modules seen in the figure. By clicking the button `Visualization panel` (located at the bottom left of the panel), the so-called visualization panel appears. This panel allows to load up to eight input images and provides several interactive tools of visualization and image manipulation such as denoising, contrast enhancement, zooming, etc.

Fig. A.2 shows the visualization panel after loading three input images and having clicked the `Fuse` button in the control panel with the default parameters (see Fig. A.1).

---

[1] The contributors to the software are: N. Gupta, H. Heijmans, A. Kumar, G. Piella, A. Steenbeek and P. de Zeeuw.

[2] Contact person: Paul de Zeeuw, e-mail: Paul.de.Zeeuw@cwi.nl



Type: Wavelet toolbox

Filter Family: Daub

Filter: Haar

Level: 5

Choose the Activity measure.

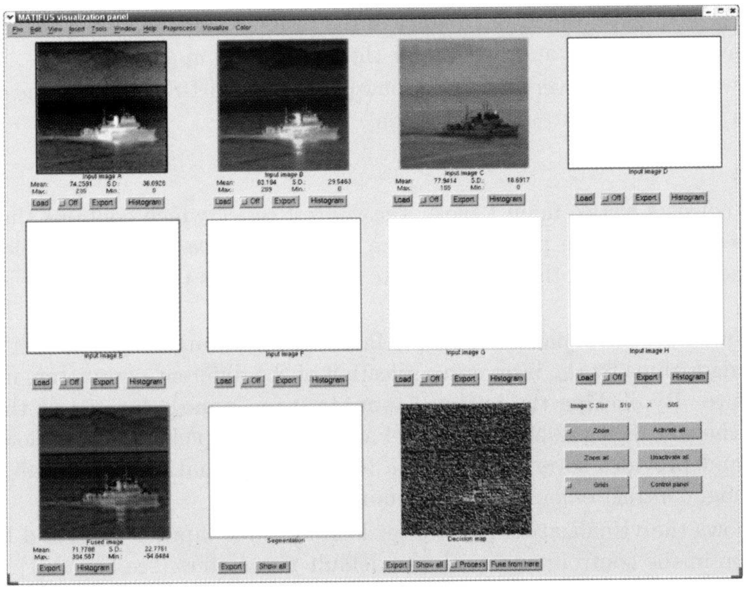
Choose the Match Measure

Approximation coefficients

Detail coefficients

Click to edit the app. and det. coefficients.

Figure A.1: *Control panel with default parameters.*



Input image A
Mean: 74.2591 S.D.: 36.0926 Max: 235 Min: 0

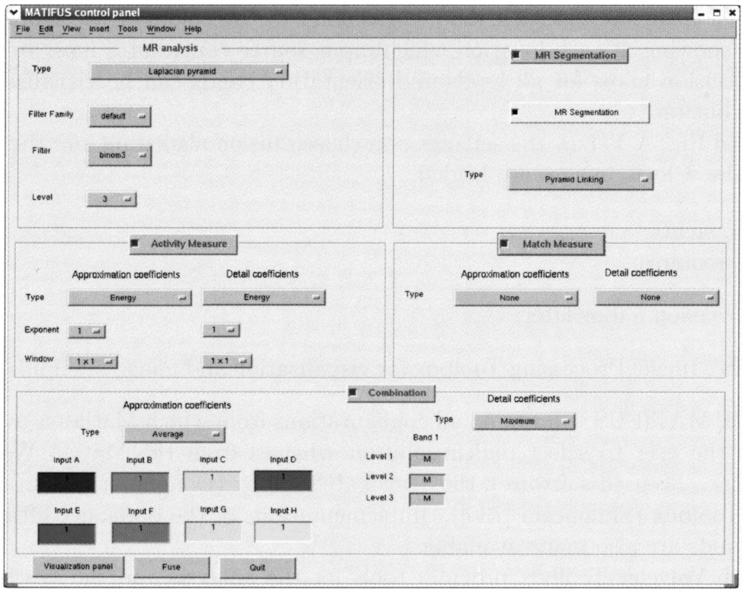
Input image B
Mean: 63.194 S.D.: 29.5463 Max: 209 Min: 0

Input image C
Mean: 77.9414 S.D.: 18.6917 Max: 155 Min: 0

Image C Size 510 × 505

Fused image
Mean: 71.7788 S.D.: 22.7761 Max: 304.587 Min: -54.8484

Figure A.2: *Visualization panel: input images (top) and corresponding composite image (bottom left) and decision map (bottom right).*



MR analysis
Type: Laplacian pyramid
Filter Family: default
Filter: binom3
Level: 3

MR Segmentation
MR Segmentation
Type: Pyramid Linking

Activity Measure

	Approximation coefficients	Detail coefficients
Type	Energy	Energy
Exponent	1	1
Window	1 x 1	1 x 1

Match Measure

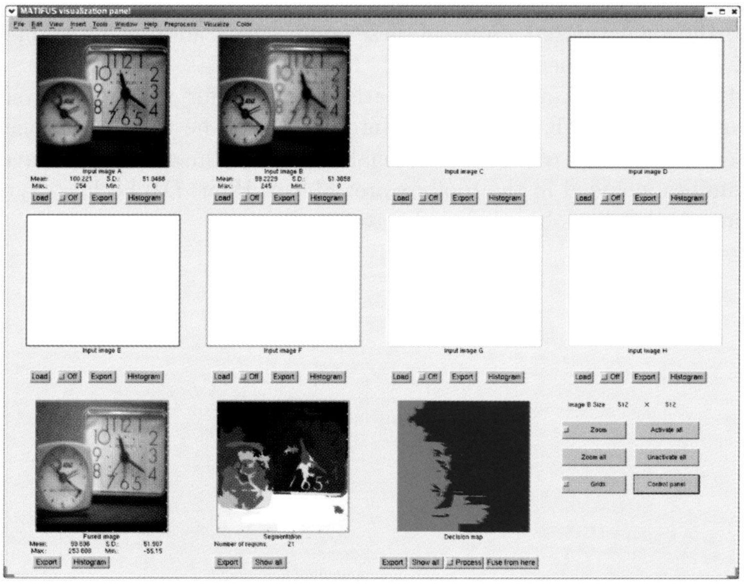
	Approximation coefficients	Detail coefficients
Type	None	None

Combination
Approximation coefficients
Type: Average

Input A	Input B	Input C	Input D
1	1	1	1
Input E	**Input F**	**Input G**	**Input H**
1	1	1	1

Detail coefficients
Type: Maximum
Band 1
Level 1: M
Level 2: M
Level 3: M

Figure A.3: *Example of an specific fusion algorithm.*



Input image A
Mean: 100.221 S.D.: 51.0468
Max.: 254 Min.: 0

Input image B
Mean: 99.2229 S.D.: 51.3058
Max.: 245 Min.: 0

Image B Size 512 x 512

Fused image
Mean: 99.636 S.D.: 51.907
Max: 253.608 Min.: -55.15

Segmentation
Number of regions: 21

Decision map

Figure A.4: *Input images and corresponding composite image, segmentation and decision map using the specific fusion algorithm of Fig. A.3.*

The colors used in the 'decision map' correspond to the colors of the borders around the input images, showing at each location which input source contributes most to the composite image. The decision maps for all levels and orientation bands can be visualized by selecting the `Show all` button.

Fig. A.3 and Fig. A.4 show the settings of a chosen fusion algorithm and the corresponding results when the 'Clock' images are loaded.

Systems requirements

MATIFUS requires

- Matlab, version 5.0 or later;
- the Matlab Image Processing Toolbox for visualization and image manipulation functions.

In principle, MATIFUS will run on all configurations from which Matlab is available. MATIFUS enables the user to select multiresolution schemes from the Matlab Wavelet Toolbox (MathWorks Inc., licensed software), the WaveLab Toolbox (Donoho *et al.* [52]), and the Matlab Pyramid Toolbox (Simoncelli [136]). Implementations of the quincunx lifting scheme and gradient pyramids are also made available.

The Matlab Wavelet Toolbox provides tools for the analysis and the synthesis of signals and images using wavelets. The two-dimensional wavelets are obtained by tensor products of one-dimensional wavelets. The transforms are not limited to dyadic sizes. The wavelet families included in the toolbox are: Morlet, Mexican hat, Meyer, Haar, Daubechies, Symlets, Coiflets and Splines biorthogonal wavelets. Some of these families have additional parameters specifying the number of vanishing moments.

The WaveLab Toolbox is a library of Matlab routines for wavelet analysis and synthesis, that is available free of charge over the Internet [52]. The two-dimensional wavelets are obtained by tensor products of one-dimensional wavelets. Images should have dyadic sizes. The wavelet families included in the toolbox are: Meyer, Haar, Daubechies, Symlets, Coiflets, Dubuc-Deslauries and Splines biorthogonal wavelets.

Bibliography

[1] A. Rosenfeld (ed.). *Multiresolution Image Processing and Analysis.* Springer-Verlag, Berlin, 1984.

[2] Abidi, M. A., and Gonzalez, R. C., Eds. *Data Fusion in Robotics and Machine Intelligence.* Academic Press, Boston, 1992.

[3] Akerman III, A. Pyramids techniques for multisensor fusion. In *Sensor Fusion V* (Boston, Massachussets, November 15-20 1992), Proceedings of SPIE, vol. 1828, pp. 124–131.

[4] Atkinson, K. E. *An Introduction to Numerical Analysis*, second ed. John Wiley and Sons, New York, 1989.

[5] Barnsley, M. F. *Fractals Everywhere.* Academic Press, San Diego, 1988.

[6] Bastiere, A. Methods for multisensor classification of airborne targets integrating evidence theory. *Aeroespace Science and Technology 2*, 6 (1998), 401–411.

[7] Beckerman, M., and Sweeney, F. J. Segmentation and cooperative fusion of laser radar image data. In *Sensor Fusion and Aerospace Applications II* (Orlando, Florida, April 6-7 1994), Proceedings of SPIE, vol. 2233, pp. 88–98.

[8] Béthune, S. d., Muller, F., and Binard, M. Adaptive intensity matching filters: a new tool for multi-resolution data fusion. In *Proceedings of the Sensor and Propagation Panel's 7th Symposium on Multi-Sensor Systems and Data Fusion for Telecommunications, Remote Sensing and Radar* (Lisbon, Portugal, 29 September–2 October 1997), vol. 28, pp. 1–15.

[9] Bister, M., Cornelis, J., and Rosenfeld, A. A critical view of pyramid segmentation algorithms. *Pattern Recognition Letters 11* (1990), 605–617.

[10] Borghys, D., Verlinde, P., Perneel, C., and Acheroy, M. Multilevel data fusion for the detection of targets using multispectral image sequences. *Optical Engineering 37*, 2 (1998), 477–484.

[11] Bracewell, R. N. *The Fourier Transform and its Applications*, second ed. McGraw-Hill, New York, 1986.

[12] Broussard, R. P., Rogers, S. K., Oxley, M. E., and Tarr, G. L. Physiologically motivated image fusion for object detection using a pulse coupled neural network. *IEEE Transactions on Neural Networks 10*, 3 (1999), 554–563.

[13] Brown, L. G. A survey of image registration techniques. *ACM Computing Surveys 24*, 4 (December 1992), 325–376.

[14] 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.

[15] Buccigrossi, R. W., and Simoncelli, E. P. Image compression via joint statistical characterization in the wavelet domain. *IEEE Transactions on Image Processing 8*, 12 (December 1999), 1688–1701.

[16] Burt, P. J. Fast filter transforms for image processing. *Computer Vision, Graphics, and Image Processing 16* (1981), 20–51.

[17] Burt, P. J. The pyramid as a structure for efficient computation. In *Multiresolution Image Processing and Analysis*, A. Rosenfeld, Ed. Springer-Verlag, Berlin, 1984, pp. 6–35.

[18] Burt, P. J. A gradient pyramid basis for pattern selective image fusion. *SID Technical Digest 16* (1985), 467–470.

[19] Burt, P. J., and Adelson, E. H. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications 31*, 4 (April 1983), 532–540.

[20] Burt, P. J., Hong, T. H., and Rosenfeld, A. Segmentation and estimation of image region properties through cooperative hierarchical computation. *IEEE Transactions on Systems, Man, and Cybernetics 11*, 12 (December 1981), 802–809.

[21] Burt, P. J., and Kolczynski, R. J. Enhanced image capture through fusion. In *Proceedings of the 4th International Conference on Computer Vision* (Berlin, Germany, May 1993), pp. 173–182.

[22] Calderbank, A. R., Daubechies, I., Sweldens, W., and Yeo, B.-L. Wavelet transforms that map integers to integers. *Applied and Computational Harmonic Analysis 5* (1998), 332–369.

[23] Candès, E. J., and Donoho, D. L. Curvelets: a surprisingly nonadaptive representation for objects with edges. In *Proceedings of the 4th International Conference on Curves and Surfaces* (Saint-Malo, France, July 1-7 1999).

[24] Candès, E. J., and Guo, F. New multiscale transforms, minimum total variation synthesis: applications to edge-preserving image reconstruction. *Signal Processing*, 82 (2002), 1519–1543.

[25] Castellanos, A., and Tardos, J. D. *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach.* Kluwer Academic Publishers, Boston, 2000.

[26] Chan, T. F., and Zhou, H. M. ENO-Wavelet transforms for piecewise smooth functions. *SIAM Journal on Numerical Analysis 40*, 4 (2002), 1369–1404.

[27] Chipman, L. J., and Orr, T. M. Wavelets and image fusion. In *Proceedings of the IEEE International Conference on Image Processing* (Washington D.C., October 24-26 1995), pp. 248–251.

[28] Cibulskis, J. M., and Dyer, C. R. An analysis of node linking in overlapped pyramids. *IEEE Transactions on Systems, Man, and Cybernetics 14* (May/June 1984), 424–436.

[29] Claypoole, R. L., Baraniuk, R. G., and Nowak, R. D. Adaptive wavelet transforms via lifting. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (Seattle, Washington, May 12-15, 1998), vol. 3, pp. 1513–1516.

[30] 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* (Asilomar, California, 1997), vol. 1, pp. 662–667.

[31] Cohen, A., and Matei, B. Compact representation of images by edge adapted multiscale transforms. In *Proceeding of the IEEE International Conference on Image Processing* (Thessaloniki, Greece, October 7-10 2001).

[32] Cohen, I., Raz, S., and Malah, D. Orthonormal shift-invariant adaptive local trigonometric decomposition. *Signal Processing 57*, 1 (February 1997), 43–64.

[33] Coifman, R. R., and Meyer, Y. Remarques sur l'analyse de Fourier à fenêtre. *Comptes Rendus de l'Académie des Sciences 312* (1991), 259–261.

[34] Coifman, R. R., Meyer, Y., and Wickerhouser, M. V. Wavelet analysis and signal processing. In *Wavelet and their Applications*. Jones and Barlett, Boston, 1992, pp. 153–178.

[35] Costantini, M., Farina, A., and Zirilli, F. The fusion of different resolution SAR images. *Proceedings of the IEEE 81*, 1 (1997), 139–146.

[36] Couloigner, I., Ranchin, T., Valtonen, V. P., and Wald, L. Benefit of the future SPOT-5 and of data fusion to urban roads mapping. *International Journal of Remote Sensing 19*, 8 (1998), 1519–1532.

[37] Cover, T. M., and Thomas, J. A. *Elements of Information Theory*. John Wiley and Sons, New York, 1991.

[38] Croisier, A., Esteban, D., and Galand, C. Perfect channel splitting by use of interpolation/decimation/tree decomposition techniques. In *Proceedings of the International Conference on Information Science and Systems* (Patras, Greece, August 1976), pp. 443–446.

[39] Crowley, J. L. *A Representation for Visual Information*. PhD thesis, Carnegie-Mellon University, Robotics Institute, Pittsburgh, Pennsylvania, 1981.

[40] Cvetković, Z., and Vetterli, M. Oversampled filter banks. *IEEE Transactions on Signal Processing 46*, 5 (May 1998), 1245–1254.

[41] Dasarathy, B. V. *Decision Fusion*. IEEE Computer Society Press, Los Alamitos, California, 1994.

[42] Dasarathy, B. V. Fuzzy evidential reasoning approach to target identity and state fusion in multisensor environments. *Optical Engineering 36*, 3 (1997), 683–699.

[43] Daubechies, I. The wavelet transform, time-frequency localization, and signal analysis. *IEEE Transactions on Information Theory 36* (1990), 961–1005.

[44] Daubechies, I. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1992.

[45] Daubechies, I., and Sweldens, W. Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications 4*, 3 (1998), 245–267.

[46] 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*, 2 (February 1998), 246–252.

[47] DeVore, R. A., Jawerth, B., and Lucier, B. J. Image compression through wavelet transform coding. *IEEE Transactions on Information Theory 38* (March 1992), 719–746.

[48] Do, M. N., and Vetterli, M. Orthonormal finite ridgelet transform for image compression. In *Proceedings of the IEEE International Conference on Image Processing* (Vancouver, Canada, September 10-13 2000).

[49] Do, M. N., and Vetterli, M. Framing pyramids. *IEEE Transactions on Signal Processing 51*, 9 (September 2003), 2329–2343.

[50] Donoho, D. L. Wedgelets: nearly minimax estimation of edges. Technical report, Statistics Department, Stanford University, California, 1997.

[51] Donoho, D. L. Orthonormal ridgelets and linear singularities. *SIAM Journal of Mathematical Analysis 31*, 5 (2000), 1062–1099.

[52] Donoho, D. L., Duncan, M., Huo, X., and Levi-Tsabari, O. Wavelab Toolbox.
Available at http://www-stat.stanford.edu/~wavelab.

[53] Dragotti, P. L., and Vetterli, M. Footprints and edgeprints for image denoising and compression. In *Proceedings of the IEEE International Conference on Image Processing* (Thessaloniki, Greece, October 7-10 2001).

[54] DUFFIN, R. J., AND SCHAEFFER, A. C. A class of nonharmonic Fourier series. *Transactions of the American Mathematical Society 72* (1952), 341–366.

[55] 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.

[56] FECHNER, T., AND GODLEWSKI, G. Optimal fusion of TV and infrared images using artificial neural networks. In *Applications and Science of Artificial Neural Networks* (Orlando, Florida, April 17-21 1995), Proceedings of SPIE, vol. 2492, pp. 919–925.

[57] 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, November 13-16 1994), vol. 2, pp. 331–335.

[58] FOURIER, J. B. *Théorie Analytique de la Chaleur.* Firmin Didot, Paris, 1822.

[59] GABOR, D. Theory of communication. *Journal of IEE 93* (1946), 429–457.

[60] GEREK, Ö. N., AND ÇETIN, A. E. Image coding using adaptive subband decomposition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (Seattle, Washington, May 12-15 1998).

[61] GEREK, Ö. N., AND ÇETIN, A. E. Adaptive polyphase subband decomposition structures for image compression. *IEEE Transactions on Image Processing 9* (October 2000), 1649–1659.

[62] GERONIMO, S., HARDIN, D. P., AND MASSOPUST, P. R. Fractal functions and wavelet expansions based on several functions. *Journal of Approximation Theory 78*, 3 (1994), 373–401.

[63] GOUTSIAS, J., AND HEIJMANS, H. J. A. M. Nonlinear multiresolution signal decomposition schemes. Part I: Morphological pyramids. *IEEE Transactions on Image Processing 9*, 11 (2000), 1862–1876.

[64] HACKBUSH, W. *Multi-Grid Methods and Applications.* Springer-Verlag, Berlin, 1985.

[65] 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.

[66] HAMPSON, F. J., AND PESQUET, J.-C. M-band nonlinear subband decompositions with perfect reconstruction. *IEEE Transactions on Image Processing 7* (1998), 1547–1560.

[67] HARTEN, A. Discrete multiresolution analysis and generalized wavelets. *Journal of Applied Numerical Mathematics 12* (1993), 153–193.

[68] HARTEN, A., ENGQUIST, B., OSHER, S., AND CHAKRAVARTHY, S. Uniformly high order essentially non-oscillatory schemes III. *Journal of Computational Physics 71* (1987), 231–303.

[69] HAYKIN, S. *Adaptive Filter Theory*, second ed. Prentice-Hall, Cambridge, 1991.

[70] 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.

[71] HEIJMANS, H. J. A. M., PIELLA, G., PESQUET-POPESCU, B., AND ABHAYARATNE, C. Adaptive wavelets for image compression using update lifting: quantization and error analysis. In preparation.

[72] HERLEY, C., KOVAČEVIĆ, J., RAMCHANDRAN, K., AND VETTERLI, M. Tilings of time-frequency plane: Construction of arbitrary orthogonal bases and fast tilings algorithms. *IEEE Transactions on Signal Processing 41*, 12 (December 1993), 3341–3360.

[73] HERLEY, C., XIONG, Z., RAMCHANDRAN, K., AND ORCHARD, M. T. An efficient algorithm to find a jointly optimal time-frequency segmentation using time-varying filter banks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (Detroit, Michigan, May 8–12 1997), pp. 1516–1519.

[74] Hill, D., and Hawkes, D. Fusing medical images. *Image Processing 6*, 2 (1994), 22–24.

[75] Hong, T. H., and Rosenfeld, A. Compact region extraction using weighted pixel linking in a pyramid. *IEEE Transactions on Pattern Analysis and Machine Intelligence 6*, 2 (1984), 222–229.

[76] Horn, R. A., and Johnson, C. R. *Matrix Analysis*. Cambridge University Press, New York, 1985.

[77] Jacquin, A. E. Fractal image coding: A review. *Proceedings of IEEE 81*, 10 (October 1993), 1451–1465.

[78] Jeon, B., and Landgrebe, D. A. Decision fusion approach for multitemporal classification. *IEEE Transactions on Geoscience and Remote Sensing 37*, 3 (1999), 1227–1233.

[79] Kam, M., Zhu, X., and Kalata, P. Sensor fusion for mobile robot navigation. *Proceedings of the IEEE 85* (January 1997), 108–119.

[80] Klinger, A. Pattern and search statistics. In *Optimizing Methods in Statistics*. Academic Press, New York, 1971.

[81] Koenderink, J. J. The structure of images. *Biological Cybernetics 50* (1984), 363–370.

[82] Kohonen, T. *Self-Organizing Maps*. Springer-Verlag, Berlin, 1995.

[83] Koren, I., Laine, A., and Taylor, F. Image fusion using steerable dyadic wavelet transforms. In *Proceedings of the IEEE International Conference on Image Processing* (Washington D.C., October 1995), pp. 232–235.

[84] Kovačević, J., and Sweldens, W. Wavelet families of increasing order in arbitrary dimensions. *IEEE Transactions on Image Processing 9*, 3 (2000), 480–496.

[85] Kovačević, J., and Vetterli, M. Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for $\mathbb{R}^n$. *IEEE Transactions on Information Theory 38* (1992), 533–555.

[86] Kruskal, J. B. Non metric multidimensional scaling: a numerical method. *Psychometrika 29* (1964), 115–129.

[87] Lallier, E. Real-time pixel-level image fusion through adaptive weight averaging. Technical report, Royal Military College of Canada, Kingston, Ontario, 1999.

[88] Le Pennec, E., and Mallat, S. G. Image compression with geometrical wavelets. In *Proceedings of the IEEE International Conference on Image Processing* (Vancouver, Canada, September 10-13 2000).

[89] Leckie, D. G. Synergism of SAR and visible/infrared data for forest type discrimination. *Photogrammetric Engineering and Remote Sensing 56* (1990), 1237–1246.

[90] Lewis, A. S., and Knowles, G. Image compression using the 2D wavelet transform. *IEEE Transactions on Image Processing 1* (April 1992), 244–250.

[91] Li, H., Manjunath, B. S., and Mitra, S. K. A contour based approach to multisensor image registration. *IEEE Transactions on Image Processing 4*, 3 (March 1995), 320–334.

[92] Li, H., Manjunath, B. S., and Mitra, S. K. Multisensor image fusion using the wavelet transform. *Graphical Models and Image Processing 57*, 3 (May 1995), 235–245.

[93] Li, S. T., and Wang, Y. N. Multisensor image fusion using discrete multiwavelet transform. In *Proceedings of the 3rd International Conference on Visual Computing* (Mexico city, Mexico, September 2000).

[94] Liu, Z., Tsukada, K., Hanasaki, K., Ho, Y. K., and Dai, Y. P. Image fusion by using steerable pyramid. *Pattern Recognition Letters 22* (2001), 929–939.

[95] Lou, K. N., and Lin, L. G. An intelligent sensor fusion system for tool monitoring on a machining centre. *International Journal of Advanced Manufacturing Technology*, 13 (1997), 556–565.

[96] Luo, R. C., and Kay, M. G. *Multisensor Integration and Fusion for Intelligent Machines and Systems.* Ablex Publishing Corporation, 1995.

[97] Mallat, S. G. Multiresolution approximations and wavelet orthonormal bases of $L^2(\mathbb{R})$. *Transactions of the American Mathematical Society 315* (1989), 69–87.

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

[99] Mallat, S. G. *A Wavelet Tour of Signal Processing.* Academic Press, San Diego, 1998.

[100] Mallat, S. G., and Zhong, S. Characterization of signals from multiscale edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence 14* (1992), 710–732.

[101] Manduca, A. Multispectral image visualization with nonlinear projections. *IEEE Transactions on Image Processing 5*, 10 (1996), 1486–1490.

[102] Marr, D. *Vision.* W. H. Freeman and Company, New York, 1982.

[103] Matsopoulos, G. K., and Marshall, S. Application of morphological pyramids: fusion of MR and CT phantoms. *Journal of Visual Communication and Image Representation 6*, 2 (1995), 196–207.

[104] Matsopoulos, G. K., Marshall, S., and Brunt, J. N. H. Multiresolution morphological fusion of MR and CT images of the human brain. *Proceedings of the IEE on Vision, Image and Signal Processing 141*, 3 (June 1994), 137–142.

[105] Matuszewski, B. J., Shark, L.-K., Varley, M. R., and Smith, J. P. Region-based wavelet fusion of ultrasonic, radiographic and shearographyc non-destructive testing images. In *Proceedings of the 15th World Conference on Non-Destructive Testing* (Rome, October 15–21 2000).

[106] McMichael, D. W. Data fusion for vehicle-borne mine detection. In *Proceedings of EUREL Conference on Detection of Abandoned Land Mines* (Edinburgh, Scotland, October 7–9 1996), pp. 167–171.

[107] Meyer, Y. *Wavelets and Operators.* Cambridge University Press, 1992.

[108] Milisavljević, N., and Acheroy, M. An approach to the use of Bayesian rule in decision level fusion for multisensor mine detection. In *Proceedings of Physics in Signal and Image Processing Conference* (Paris, France, January 1999), pp. 261–266.

[109] Mirhosseini, A. R., Hong, Y., Kin, M. L., and Tuan, P. Human face image recognition: an evidence aggregation approach. *Computer Vision and Image Understanding 71*, 2 (1998), 213–230.

[110] Moore, B. C. J. *An Introduction to the Psychology of Hearing*, third ed. Academic Press, San Diego, 1989.

[111] Mukhopadhyay, S., and Chanda, B. Fusion of 2D grayscale images using multiscale morphology. *Pattern Recognition 34* (2001), 1939–1949.

[112] Murphy, R. R. Sensor and information fusion improved vision-based vehicle guidance. *IEEE Intelligent Systems 13*, 6 (December 1998), 49–56.

[113] Nacken, P. F. M. Image segmentation by connectivity preserving relinking in hierarchical graphs structures. *Pattern Recognition 9*, 6 (1995), 907–920.

[114] Newman, E. A., and Hartline, P. H. The infrared vision of snakes. *Scientific American 246*, 3 (1982), 116–127.

[115] Pohl, C., and Genderen, J. L. Multisensor image fusion in remote sensing: concepts, methods and applications. *International Journal of Remote Sensing 19*, 5 (1998), 823–854.

[116] Pu, T., and Ni, G. Contrast-based image fusion using the discrete wavelet transform. *Optical Engineering 39*, 8 (August 2000), 2075–2082.

[117] Qu, G. H., Zhang, D. L., and Yan, P. F. Information measure for performance of image fusion. *Electronic Letters 38*, 7 (2002), 313–315.

[118] Ramchandran, K., Xiong, Z., Asai, K., and Vetterli, M. Adaptive transforms for image coding using spatially varying wavelet packets. *IEEE Transactions on Image Processing 5*, 7 (July 1996), 1197–1204.

[119] Ranchin, T., and Wald, L. The wavelet transform for the analysis of remotely sensed images. *International Journal of Remote Sensing 14* (1993), 615–619.

[120] Ranchin, T., and Wald, L. Fusion of high spatial and spectral resolution images: the ARSIS concept and its implementation. *Photogrammetric Engineering and Remote Sensing 66* (2000), 49–61.

[121] Reed, J. M., and Hutchinson, S. Image fusion and subpixel parameter estimation for automated optical inspection of electronic components. *IEEE Transactions on Industrial Electronics 43*, 3 (1996), 346–354.

[122] Richards, J. A. Thematic mapping from multitemporal image data using the principal component transformation. *Remote Sensing of Environment 16* (1984), 36–46.

[123] Rockinger, O. Pixel-level fusion of image sequences using wavelet frames. In *Proceedings of the 16th Leeds Annual Statistical Research Workshop* (1996), Leeds University Press, pp. 149–154.

[124] Ryan, D., and Tinkler, R. Night pilotage assessment of image fusion. In *Helmet and Head-Mounted Displays and Symbology Design Requirements II* (Orlando, Florida, April 18-19 1995), Proceedings of SPIE, vol. 2465, pp. 50–67.

[125] Said, A., and Pearlman, W. A. An image multiresolution representation for lossless and lossy compression. *IEEE Transactions on Image Processing 5* (1996), 1303–1310.

[126] Sammon, J. W. A nonlinear mapping for data analysis. *IEEE Transactions on Computers C-18* (1969), 401–409.

[127] Sayood, K. *Introduction to Data Compression*, second ed. Morgan Kaufmann Publishers, San Francisco, 2000.

[128] Scheunders, P. Multiscale edge representation applied to image fusion. In *Wavelet Applications in Signal and Image Processing VIII* (San Diego, California, July 30-August 4 2000), Proceedings of SPIE, vol. 4119.

[129] Scheunders, P. An orthogonal wavelet representation of multivalued images. *IEEE Transactions on Image Processing 12*, 6 (June 2003), 718–725.

[130] Scheunders, P., and De Backer, S. Fusion and merging of multispectral images using multiscale fundamental forms. *Journal of the Optical Society of America A 18*, 10 (2001), 2468–2477.

[131] Serra, J. *Image Analysis and Mathematical Morphology*, vol. 1. Academic Press, New York, 1982.

[132] Shannon, C. E. A mathematical theory of communication. *Bell System Technical Journal 27* (1948), 379–423, 623–656.

[133] Shannon, C. E. Communications in the presence of noise. *Proceedings of the IRE 37* (January 1949), 10–21.

[134] Shapiro, J. Embedded image coding using zerotrees of wavelets coefficients. *IEEE Transactions on Signal Processing 41*, 12 (December 1993), 3445–3462.

[135] Sharma, R. K., and Pavel, M. Adaptive and statistical image fusion. *Society for Information Display Digest of Technical Papers 27* (1996), 969–972.

[136] Simoncelli, E. P. Matlab Pyramid Toolbox.
Available at ftp://ftp.cis.upenn.edu/pub/eero/matlabPyrTools.tar.gz.

[137] Simoncelli, E. P., and Freeman, W. T. The steerable pyramid: a flexible architecture for multi-scale derivative computation. In *Proceedings of the IEEE International Conference on Image Processing* (Washington, DC., October 23-26 1995), pp. 444–447.

[138] Simoncelli, E. P., Freeman, W. T., Adelson, E. H., and Heeger, D. J. Shiftable multi-scale transforms. *IEEE Transactions on Information Theory 38*, 2 (1992), 587–607.

[139] Smith, M. J. T., and Barnwell III, T. P. Exact reconstruction for tree structured subband coders. *IEEE Transactions on Acoustics, Speech, and Signal Processing 34* (June 1986), 434–441.

[140] Soille, P. *Morphological Image Analysis.* Springer-Verlag, Berlin, 1999.

[141] Spann, M., and Horne, C. Image segmentation using a dynamic thresholding pyramid. *Pattern Recognition*, 22 (1989), 719–732.

[142] Starck, J.-L., Candès, E. J., and Donoho, D. L. The curvelet transform for image denoising. *IEEE Transactions on Image Processing 11*, 6 (June 2002), 670–684.

[143] Strang, G., and Nguyen, T. *Wavelets and Filter Banks.* Wellesley-Cambridge Press, Wellesley, Massachusetts, 1996.

[144] Strela, V., Heller, N., and Strang, G. The applications of multiwavelets filter banks to signal and image processing. *IEEE Transactions on Image Processing 8*, 4 (1996), 548–563.

[145] Strela, V., and Strang, G. Finite element multiwavelets. In *Approximation Wavelets, Theory and Applications.* Kluwer Academic Publishers, Boston, 1995, pp. 485–496.

[146] Sweldens, W. The lifting scheme: A new philosophy in biorthogonal wavelet constructions. In *Wavelet Applications in Signal and Image Processing III* (San Diego, California, July 12-14 1995), Proceedings of SPIE, vol. 2569, pp. 68–79.

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

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

[149] Sworder, D. D., Boyd, J. E., and Clapp, G. A. Image fusion for tracking manoeuvring targets. *International Journal of Systems Science 28*, 1 (1997), 1–14.

[150] Taubman, D. S., and Marcellin, M. W. *JPEG2000: Image Compression Fundamentals, Standards, and Practice.* Kluwer Academic Publishers, Boston, 2002.

[151] Thévenaz, P., and Unser, M. An efficient mutual information optimizer for multiresolution image registration. In *Proceedings of the IEEE International Conference on Image Processing* (Chicago, Illinois, October 4-7 1998), vol. 1, pp. 833–837.

[152] Toet, A. Image fusion by a ratio of low-pass pyramid. *Pattern Recognition 9* (1989), 245–253.

[153] Toet, A. A morphological pyramidal image decomposition. *Pattern Recognition Letters 9* (1989), 255–261.

[154] Toet, A. Hierarchical image fusion. *Machine Vision Application* (March 1990), 1–11.

[155] Toet, A. Multiscale contrast enhancement with application to image fusion. *Optical Engineering 31*, 5 (1992), 1026–1031.

[156] Toet, A., and Franken, E. M. Perceptual evaluation of different image fusion schemes. *Displays 24*, 1 (February 2003), 25–37.

[157] Toet, A., IJspeert, J. K., Waxman, A. M., and Aguilar, M. Fusion of visible and thermal imagery improves situational awareness. *Displays 18*, 2 (December 1997), 85–95.

[158] Toet, A., and Walraven, J. New false color mapping for image fusion. *Optical Engineering 35*, 3 (1996), 650–658.

[159] Toutin, T. SPOT and Landsat stereo fusion for data extraction over mountainous areas. *Photogrammetric Engineering and Remote Sensing 64*, 2 (1998), 109–113.

[160] Trappe, W., and Liu, K. J. R. Adaptivity in the lifting scheme. In *Proceedings of the 33rd Annual Conference on Information Sciences and Systems* (Baltimore, Maryland, March 17-19 1999), pp. 950–955.

[161] van Elsen, P. A., Pol, E. J. D., and Viergever, M. A. Medical image matching – A review with classification. *IEEE Transactions on Engineering Medical Biology 12*, 1 (March 1993), 26–39.

[162] Vetterli, M. Filter banks allowing perfect reconstruction. *Signal Processing 10*, 6 (1986), 219–244.

[163] Vetterli, M., and Kovačević, J. *Wavelets and Subband Coding.* Prentice Hall, Englewood Cliffs, New Jersey, 1995.

[164] Vincken, K. L., Koster, A. S. E., and Vierger, M. A. Probabilistic multiscale image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*, 2 (1997), 109–120.

[165] Wald, L., Ranchin, T., and Mangolini, M. Fusion of satellite images of different spatial resolutions: assessing the quality of resulting images. *Photogrammetric Engineering and Remote Sensing 63*, 3 (1997), 691–699.

[166] Wang, Z., and Bovik, A. C. A universal image quality index. *IEEE Signal Processing Letters 9*, 3 (March 2002), 81–84.

[167] Waxman, A. M., Gove, A. N., Fay, D. A., Racamato, J. P., Carrick, J. E., Seibert, M. C., and Savoye, E. D. Color night vision: opponent processing in the fusion of visible and IR imagery. *Neural Networks 10*, 1 (1997), 1–6.

[168] Wickerhauser, V. M., and Coifman, R. R. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory 38* (1992), 713–718.

[169] Wilson, T. A., Rogers, S. K., and Meyers, L. R. Perceptual based hyperspectral image fusion using multiresolution analysis. *Optical Engineering 34*, 11 (1995), 3154–3164.

[170] Witkin, A. P. Scale-space filtering. In *Proceedings of the 8th International Joint Conference of Artificial Intelligence* (Karlsruhe, Germany, Augustus 1983), pp. 1019–1022.

[171] Wong, S. T. C., Knowlton, R. C., Hawkins, R. A., and Laxer, K. D. Multimodal image fusion for noninvasive epilepsy surgery planning. *IEEE Transactions on Computer Graphics and Applications 16*, 1 (1996), 30–38.

[172] Xydeas, C., and Petrović, V. Objective pixel-level image fusion performance measure. In *Sensor Fusion: Architectures, Algorithms and Applications IV* (Orlando,Florida, April 24-28 2000), Proceedings of SPIE, vol. 4051, pp. 88–99.

[173] Zhang, Z. *Investigations of Image Fusion.* PhD thesis, Lehigh University, Bethlehem, Philadelphia, April 1999.

[174] Zhang, Z., and Blum, R. A region-based image fusion scheme for concealed weapon detection. In *Proceedings of the 31st Annual Conference on Information Sciences and Systems* (Baltimore, Maryland, March 1997), pp. 168–173.

[175] Zhang, Z., and Blum, R. A categorization of multiscale-decomposition-based image fusion schemes with a performance study for a digital camera application. *Proceedings of the IEEE 87*, 8 (August 1999), 1315–1326.

Index

List of publications

Journal papers

1. Heijmans, H. J. A. M., Pesquet-Popescu, B., and Piella, G. Building nonredundant adaptive wavelets by update lifting. (Based on technical report 17, see below.) Submitted to *Applied and Computational Harmonic Analysis*, 2003.
2. Piella, G. A general framework for multiresolution image fusion: from pixels to regions. (Based on technical report 16, see below.) To appear in *Information Fusion*, 2003.
3. Piella, G., Pesquet-Popescu, B., and Heijmans, H. J. A. M. Adaptive update lifting with a decision rule based on derivative filters. *IEEE Signal Processing Letters 9*, 10 (October 2002), 329–332.
4. Piella, G., and Heijmans, H. J. A. M. Adaptive lifting schemes with perfect reconstruction. *IEEE Transactions on Signal Processing 50*, 7 (July 2002), 1620–1630.

Conference papers

5. Abhayaratne, C., Piella, G., Pesquet-Popescu, B., and Heijmans, H. J. A. M. Adaptive integer to integer wavelet transforms using update lifting. In *Wavelets X. Proceedings of SPIE* (San Diego, California, Augustus 3–8 2003).
6. Pesquet-Popescu, B., Heijmans, H. J. A. M., Abhayaratne, C., and Piella, G. Quantization of adaptive 2D wavelet decompositions. In *Proceedings of the IEEE International Conference on Image Processing* (Barcelona, Spain, September 14–17 2003).
7. Piella, G., and Heijmans, H. J. A. M. A new quality metric for image fusion. In *Proceedings of the IEEE International Conference on Image Processing* (Barcelona, Spain, September 14–17 2003).
8. Heijmans, H. J. A. M., Piella, G., and Pesquet-Popescu, B. Building adaptive 2D wavelet decompositions by update lifting. In *Proceedings of the IEEE International Conference on Image Processing* (Rochester, New York, September 22–25 2002).
9. Piella, G., and Heijmans, H. J. A. M. Multiresolution image fusion guided by a multimodal segmentation. In *Advanced Concepts for Intelligent Vision Systems* (Ghent, Belgium, September 9–11 2002), pp. 175–182.
10. Pesquet-Popescu, B., Piella, G., and Heijmans, H. J. A. M. Construction of adaptive wavelets using update lifting and quadratic decision criteria. In *Proceedings of the European Signal Processing Conference (EUSIPCO)* (Toulose, France, September 3–6 2002).

11. Piella, G. A region-based multiresolution image fusion algorithm. In *Proceedings of the 5th International Conference on Information Fusion* (Annapolis, Maryland, July 8–11 2002), pp. 1557–1564. Award for best student paper.
12. Pesquet-Popescu, B., Piella, G., and Heijmans, H. J. A. M. Adaptive update lifting with gradient criteria modeling high-order differences. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (Orlando, Florida, May 13–17 2002).
13. Piella, G., Heijmans, H. J. A. M., and Pesquet-Popescu, B. Adaptive wavelet decompositions driven by the gradient norm. In *Proceedings of the 3rd IEEE Benelux Signal Processing Symposium* (Leuven, Belgium, March 21–22 2002), pp. 109–112.
14. Piella, G., and Heijmans, H. J. A. M. An adaptive update lifting scheme with perfect reconstruction. In *Proceedings of the IEEE International Conference on Image Processing* (Thessaloniki, Greece, October 7–10 2001), pp. 190–193.

Others

15. Heijmans, H. J. A. M., Piella, G., Pesquet-Popescu, B., and Abhayaratne, C. Adaptive wavelets for image compression using update lifting: quantization and error analysis. In preparation.
16. Piella, G. A general framework for multiresolution image fusion: from pixels to regions. Research Report PNA-R0211, CWI, Amsterdam, May 2002.
17. Heijmans, H. J. A. M., Pesquet-Popescu, B., and Piella, G. Building nonredundant adaptive wavelets by update lifting. Research Report PNA-R0212, CWI, Amsterdam, May 2002.
18. Heijmans, H. J. A. M., Kamstra, L., Piella, G., and Goutsias, J. Building nonlinear wavelets. In *Fourth Quinquennial Review 1996–2001*. Dutch Society for Pattern Recognition and Image Processing, 2001, pp. 381–398.

Samenvatting

Het begrip 'beeldinhoud' is niet eenduidig: het is afhankelijk van het resolutieniveau waarop een beeld wordt beschouwd. Deze eigenschap van beelden heeft geleid tot een belangrijk paradigma in beeldverwerking en in computer vision: om een beeld goed te begrijpen, is het belangrijk om het op een groot aantal resolutieniveaus te analyseren. Dit leidt tot de zogenaamde multiresolutie-technieken, die in vele soorten en maten in de literatuur terug zijn te vinden. Voorbeelden hiervan zijn quadtrees, pyramides, fractals, scale-spaces en wavelets.

Wavelets hebben een grote invloed gehad op signaalverwerking, zowel in het een- als in het meerdimensionale geval, en niet in de laatste plaats vanwege hun succesvolle inzet bij verschillende toepassingen. De toepasbaarheid van de wavelet transformatie en andere multiresolutie decomposities wordt echter enigzins beperkt wanneer uitsluitend lineaire operatoren worden beschouwd. De representatie en/of approximatie van een signaal door middel van een lineaire operator is in veel gevallen niet compatibel met kenmerken of eigenschappen van dat signaal. Denk bijvoorbeeld aan het effect dat een Gaussisch filter heeft op de randen van een beeld. In zulke gevallen is het gebruik van lineaire procedures niet de meest voor de hand liggende keuze.

Meer in het bijzonder geldt dat klassieke lineaire wavelets vaak niet geschikt zijn voor hoger-dimensionale signalen aangezien zulke wavelets niet goed omgaan met de 'vorm' van hoger dimensionale singulariteiten in signalen. Wanneer een beeld bijvoorbeeld bestaat uit homogene gebieden gescheiden door stuksgewijs gladde krommen, onderscheiden wavelets wel de discontinuïteit in de dwarsrichting op de krommen, maar nemen ze de gladheid in de lengterichting van de kromme niet goed waar. Deze eigenschappen illustreren de behoefte aan multiresolutie representaties van signalen die data-afhankelijk zijn.

Het belang van zulke data-gestuurde of adaptieve representaties is door verschillende onderzoekers herkend, en heeft geleid tot een overweldigende hoeveelheid nieuwe aanpakken in de wavelet-theorie, zoals bandelets, ridgelets, curvelets en morfologische wavelets. Deze technieken hebben met elkaar gemeen dat ze proberen om bestaande ideeën uit de multiresolutie-analyse te verrijken met geometrische concepten.

Dit proefschrift onderzoekt de constructie van *adaptieve wavelets* door middel van een uitbreiding van het lifting schema. Het klassieke lifting schema, in eerste instantie ontwikkeld door Wim Sweldens in 1995 [146,147], maakt het mogelijk om, uitgaande van bestaande wavelets, op een eenvoudige maar flexibele manier nieuwe, mogelijk niet-lineaire wavelets te construeren. Het schema bestaat uit het aanpassen van een bestaande wavelet-transformatie door middel van een aantal predictie en update stappen. Een update stap berekent een aanpassing van de approximatie-band, waarbij slechts gebruik wordt gemaakt van informatie in de detail-band. Hierdoor kan de bewerking omgekeerd worden en is de perfecte reconstructie van het oorspronkelijke signaal gegarandeerd. Het proefschrift stelt een adaptief lifting schema voor dat bestaat uit een adaptieve update stap, mogelijk gevolgd door een of meer vaste (predictie en update) lifting stappen. De update stap is adaptief in de zin dat er gekozen kan worden uit verschillende update filters. De keuze voor een bepaald filter wordt gemaakt aan de hand van een zogenaamde 'decision map' die afhankelijk is van lokale informatie in de input; denk bijvoorbeeld aan de gradiënt van het oorspronkelijke signaal. Op deze manier is het mogelijk om slechts de homogene gebieden te filteren terwijl discontinuïteiten behouden blijven. Een extra eis die we aan onze aanpak stellen is dat er voor de decision map geen boekhouding nodig is. We willen deze map kunnen reconstrueren aan de decoder kant. Dit is wiskundig verre van triviaal. In hoofdstuk 3 en 4 worden noodzakelijke en voldoende voorwaarden afgeleid voor verschillende beslissingsscenario's. Hoofdstuk 5 laat zien dat zulke adaptieve schema's vaak decomposities opleveren die resulteren in een lagere signaalentropie dan schema's met vaste update filters, een eigenschap die erg van pas komt bij *signaalcompressie*. Hoofdstuk 5 gaat ook kort in op het gedrag van adaptieve schema in combinatie met kwantisatie.

Het proefschift bestudeert verder beeldfusie (het combineren van beelden), een ander toepassingsgebied waar wavelets ingezet worden. In het afgelopen decennium zijn verschillende algoritmen voor beeldfusie ontwikkeld en beschreven in de literatuur. Velen zijn gebaseerd op multiresolutie-technieken zoals pyramides en wavelets. Het is gebleken dat deze systemen robuust zijn en een hoge kwaliteit beelden opleveren. Het basisidee van een typische multiresolutie beeldfusie-techniek is om de multiresolutie representaties van de verschillende input beelden te combineren in een nieuwe representatie. Het fusiebeeld kan dan worden verkregen door de inverse multiresolutie-transformatie toe te passen op de aldus verkregen samengestelde multiresolutie-representatie.

In hoofdstuk 6 en 7 stellen we een axiomatisch kader voor beeldfusie op basis van multiresolutie-technieken voor. Dit kader biedt onderdak aan de meeste bestaande multiresolutie beeldfusie-technieken en laat ook nieuwe toe. In het kader van dit proefschrift hebben we een nieuwe techniek ontwikkeld die eerst, middels een segmentatie, de onderliggende beelden in gebieden opdeelt en vervolgens een samengestelde representatie maakt uitgaande van deze gebieden. Een geheel andere aanpak gaat wederom uit van adaptieve multiresolutie decomposities. Het idee hier is dat de te fuseren beelden bepalen welke multiresolutie-transformatie en welke fusie-parameters zullen worden gebruikt. Een aantal (voorlopige) resultaten gebaseerd op de adaptieve wavelets uit voorgaande hoofstukken worden gepresenteerd. Tenslotte zal in hoofdstuk 8 een aantal nieuwe objectieve kwaliteitsmaten voor beeldfusie worden geïntroduceerd. Deze maten zijn vooral interessant omdat ze geen referentie-beeld nodig hebben.

Abstract

Over the past few years, wavelets have become extremely popular in signal and image processing applications. The classical linear wavelet transform, however, performs a homogeneous smoothing of the signal contents which, in some cases, is not desirable. This has led to a growing interest in (nonlinear) wavelet representations that can preserve discontinuities, such as transitions and edges.

In this thesis, we present the construction of *adaptive wavelets* by means of an extension of the lifting scheme. The basic idea is to choose the update filters according to some decision criterion which depends on the local characteristics of the input signal. In this way, only homogeneous regions are smoothed while discontinuities are preserved. An interesting aspect of our approach is that it is neither causal nor redundant, i.e., it does not require any bookkeeping to enable perfect reconstruction. We show that these adaptive schemes yield lower entropies than schemes with fixed update filters, a property that is highly relevant in the context of *compression*.

Another main topic of this thesis is *image fusion* using wavelets and other multiresolution representations. We propose an axiomatic setup for multiresolution-based fusion which encompasses most of the existing multiresolution image fusion schemes, but also allows the construction of new ones, both pixel and region-based. The purpose of such fusion framework is to use it for the development of adaptive fusion techniques in which the source images determine the type of multiresolution transform being used as well as the parameters that are involved.

Acknowledgments

I would like to thank all people who, in one way or another, has supported me in my research and during my staying in Amsterdam.

Above all, I am deeply indebted to my supervisor Dr. ir. H. J. M. A. Heijmans, not only because his guidance made this thesis a reality, but also because he devoted his time to me whenever I needed it. His qualities both as a person and as a researcher will always serve to me as an example of the perfect supervisor. I really appreciate his patience and scientific and human encouragements. I have learned from him a lots of things ranging from doing research to writing a scientific paper. The time and effort he put in reviewing the drafts of this thesis need also to be acknowledged.

I would like to thank my promotor Prof. Dr. P. W. Hemker, who accepted me as a doctoral student and who had confidence in my work and dedication. I also want to express my gratitude to the review committee of the thesis for their time and for their constructive remarks.

Many thanks to all my colleagues in CWI for creating such a supportive and helpful environment. Specially to Lute Kamstra for being such a great office-mate all these years and for assisting me with all kind of practical problems, and to Paul de Zeeuw for his many useful suggestions as well as for his invaluable help in the implementation of the toolbox MATIFUS.

I am indebted to many people for cooperating and sharing scientific experiences. I would particularly like to thank Prof. Dr. B. Pesquet-Popescu. It has been a pleasure to work with her.

I am also indebted to STW, which financially supported my work, and all the members of the STW committee. Their fruitful discussions and suggestions have given me many helpful ideas and opened up many interesting directions for my research.

I would like to thank all my friends I have met here in the Netherlands, because even though I cannot write a word for everyone, I remember all of them. Special thanks to Cora, Begoña, Goyita, Loles, Manon, Marlies, María Angeles, María Rosa, Míriam, Pascal and other people from Aenstal for their patience, help, constant support and encouragement. Their happiness and warm friendship have given me a lot of positive energy during all these years.

I am mostly grateful to all my family for helping me to overcome the little and great problems, and for their love during all these years. Very special thanks to my brother Andreu for being such a great friend and wonderful person, and to my mother for her patience, strength and understanding. Gracias mamá, Andreu y Miquel porque siempre habéis estado cuando os he necesitado.

Amsterdam, July 2003.