

DIRECTIONLETS: ANISOTROPIC MULTI-DIRECTIONAL REPRESENTATION WITH SEPARABLE FILTERING

THÈSE N^o 3358 (2005)

PRÉSENTÉE À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

Institut de systèmes de communication

SECTION DES SYSTÈMES DE COMMUNICATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Vladan VELISAVLJEVIC

Master of Electrical Sciences, University of Belgrade, Serbie et Monténégro
et de nationalité serbe et monténégrine

acceptée sur proposition du jury:

Prof. M. Vetterli, Dr B. Beferull-Lozano, directeurs de thèse
Prof. R. Baraniuk, rapporteur
Dr P. Dragotti, rapporteur
Prof. P. Vandergheynst, rapporteur

Lausanne, EPFL
2005

Abstract

Efficient representation of geometrical information in images is very important in many image processing areas, including compression, denoising and feature extraction. However, the design of transforms that can capture these geometrical features and represent them with a sparse description is very challenging.

Recently, the separable wavelet transform achieved a great success providing a computationally simple tool and allowing for a sparse representation of images. However, in spite of the success, the efficiency of the representation is limited by the spatial isotropy of the wavelet basis functions built in the horizontal and vertical directions as well as the lack of directionality. One-dimensional discontinuities in images (edges and contours), which are very important elements in visual perception, intersect with too many wavelet basis functions leading to a non-sparse representation. To capture efficiently these anisotropic geometrical structures characterized by many more than the horizontal and vertical directions, more flexible multi-directional and anisotropic transforms are required.

We present a new lattice-based *perfect reconstruction* and *critically sampled* anisotropic multi-directional wavelet transform. The transform retains the *separable* filtering, subsampling and simplicity of computations and filter design from the standard two-dimensional wavelet transform, unlike in the case of some other existing directional transform constructions (e.g. curvelets, contourlets or edgelets). The corresponding anisotropic basis functions, which we call *directionlets*, have directional vanishing moments along *any* two directions with rational slopes.

Furthermore, we show that this novel transform provides an efficient tool for non-linear approximation of images, achieving the decay of mean-square error $O(N^{-1.55})$, which, while slower than the optimal rate $O(N^{-2})$, is much better than $O(N^{-1})$ achieved with wavelets, but at similar complexity. Owing to critical sampling, directionlets can easily be applied to image compression since it is possible to use Lagrange optimization as opposed to the case of overcomplete expansions. The compression algorithms based on directionlets outperform the methods based on the standard wavelet transform achieving better numerical results and visual quality of the reconstructed images. Moreover, we have adapted image denoising algorithms to be used in conjunction with an undecimated version of directionlets obtaining results that are competitive with the current state-of-the-art image denoising methods while having lower computational complexity.

Résumé

La représentation adéquate de l'information géométrique dans des images est très importante à nombreuses applications du traitement d'image, y compris la compression, la réduction du bruit et l'extraction de caractéristiques. Pourtant, la conception des transformations pouvant capturer ces caractéristiques géométriques et les représenter dans une description éparsée est très compliqué.

Récemment, la transformation en ondelettes séparable a connu un grand succès fournissant un outil simple et permettant une représentation éparsée des images. Pourtant, malgré le succès, l'efficacité de la représentation est limitée par l'isotropie spatiale des fonctions de base des ondelettes établie dans les directions horizontales et verticales. Les discontinuités unidimensionnelles dans les images (les bords et les contours) qui sont les éléments très importants dans la perception visuelle, intersectent trop de fonctions de base des ondelettes et conduisent à une représentation non-éparsée. Pour capturer efficacement ces structures géométriques anisotropes caractérisées par beaucoup plus que les directions horizontales et verticales, des transformations orientables et anisotropes plus complexées sont nécessaires.

Nous présentons une nouvelle transformation en ondelettes anisotropes orientables treillis-basées *reconstruction parfaite et échantillonnée critique*. La transformation maintient le filtrage et le sous-échantillonnage *séparable* et la simplicité des calculs et de la conception des filtres de la transformation en ondelettes bidimensionnelle standard, à la différence du cas de quelque autre constructions des transformations directionnelles (par exemple des contourlets ou des edgelets). Les fonctions anisotropes correspondantes de base (*directionlets*) ont des moments volatilisé directionnels le long de deux directions *quelconques* avec des pentes rationnelles.

De plus, nous démontrons que cette nouvelle transformation fournit un outil efficace pour l'approximation non linéaire des images, réalisant l'affaiblissement de l'erreur de carré moyen $O(N^{-1.55})$, qui est concurrentiel aux taux réalisés par les autres constructions des transformations *sur-échantillonnées*. En raison de l'échantillonnage critique, des directionlets sont facilement appliqués dans les algorithmes de compression d'image évitant la question d'une représentation sur-échantillonnée. Les algorithmes de compression basés sur des directionlets surpassent les méthodes basées sur la transformation en ondelettes standard réaliser de meilleurs résultats numériques et la qualité visuelle des images reconstruites. En plus, la version sur-échantillonnée des directionlets appliqués dans un algorithme de la réduction du bruit est concurrentielle aux méthodes de la réduction du bruit d'image actuels ayant la complexité des calculs inférieure.

Acknowledgments

I am deeply thankful to my advisor Martin Vetterli who introduced me to the scientific world and has been giving me a strong motivation and guidance to complete this thesis. His enthusiasm and creativeness have not only inspired my research but also have influenced my vision of life.

I also want to thank my inexhaustible co-advisor Baltasar Beferull-Lozano for a fruitful collaboration and many hours (mostly late evening) of discussion. Pier Luigi Dragotti, the first LCAV member I met, made my stay at Imperial College wonderful and has been continuously helping me in research.

I thank all my LCAV colleagues for their generous help and relaxing coffee breaks. In particular, I am thankful to Jocelyne and Williams who built up the background for my work.

A number of my Lausanne friends have made me to feel comfortable in the last five years. They have shared hectoliters of beer with me and provided me with a wonderful coffee machine, an essential factor of my thesis. My non-Lausanne friends, whom I meet at conferences and journeys around the globe, have given me many enjoyable and memorable moments.

My special thanks go to my fiancée Valentina, who is a very patient person. She knew how to handle with my variable mood going up and down following the daily outcome of my research. She certainly provided me with an indispensable support. Finally, I would like to dedicate this work to my family, whose constant understanding and encouragement have really brought me here.

Contents

Abstract	iii
Résumé	v
Acknowledgments	vii
List of Figures	xii
List of Tables	xiii
List of Algorithms	xv
1 Introduction	1
1.1 Motivation	1
1.2 Related Work	4
1.3 Thesis Outline and Contribution	5
2 Anisotropic Two-Dimensional Wavelet Decompositions	7
2.1 Introduction	7
2.2 Standard Decomposition	8
2.3 Fully separable decomposition	10
2.4 Anisotropic wavelet decomposition	12
2.5 Conclusions	15
3 Lattice-based Skewed Discrete Wavelet Transforms	17
3.1 Introduction	17
3.2 Discretization of Directions	18
3.3 Directional Interaction	18
3.4 Lattice-based Filtering and Subsampling	21
3.5 Skewed Wavelet Transforms	23
3.6 Polyphase Representation	26
3.7 Conclusions	29
4 General Multi-directional Wavelet Transforms and Regularity	31
4.1 Introduction	31
4.2 General Lattice-based Multi-directional Construction	32
4.2.1 Cyclic Construction	34

4.2.2	Unbalanced Construction	37
4.3	Regularity of Multi-directional Wavelet Bases	37
4.3.1	Two-directional Constructions	39
4.3.2	Cyclic Construction	40
4.4	Conclusions	42
5	Non-linear Approximation and Compression of Images using Directionlets	43
5.1	Introduction	43
5.2	Achievable Approximation and Compression Rate by Directionlets	44
5.3	Lagrange Optimization	48
5.3.1	Equivalent Unconstrained Solution	49
5.3.2	Finding the Optimal λ^*	51
5.4	Anisotropic Zerotree Compression Algorithm	51
5.4.1	Standard Embedded Zerotrees	52
5.4.2	Skewed Anisotropic Zerotrees	55
5.4.3	Optimized Compression Algorithm with Spatial Segmentation	59
5.5	Simulation Results	62
5.5.1	Implementation Details	64
5.5.2	Experiments	64
5.6	Conclusions	74
5.A	Proofs	74
5.A.1	Proof of Equation 5.1	74
5.A.2	Proof of Theorem 5.1	75
6	Denoising of Images using Directionlets	81
6.1	Introduction	81
6.2	Gaussian Scale Mixtures Estimation Model	82
6.2.1	Image Model	82
6.2.2	Noise Model	84
6.2.3	Estimation of Image in Noise	85
6.2.4	Image Denoising Algorithm	86
6.3	Directionlets-based Image Denoising	86
6.3.1	Isotropic and Anisotropic Neighborhoods	88
6.3.2	Smooth Denoising	90
6.3.3	Oriented Denoising	90
6.3.4	Combined Smooth and Oriented Denoising	92
6.4	Numerical Experiments	96
6.5	Conclusions	102
7	Conclusion	103
7.1	Summary	103
7.2	Future Work	104
	Bibliography	105
	Curriculum Vitae	115

List of Figures

2.1	Inefficiency of isotropic basis functions	8
2.2	The standard 2-D WT decomposition and the basis functions	9
2.3	The class <i>Mondrian</i> and the results of the anisotropic transforms	11
2.4	The FSWT decomposition and the basis functions	13
2.5	The AWT decomposition and the basis functions	14
3.1	The class <i>S-Mondrian</i> and the results of the anisotropic transforms	19
3.2	Directional interaction	20
3.3	The intersections between cosets and digital lines - co-lines	21
3.4	The lattice-based filtering and subsampling	22
3.5	The basis functions obtained by the skewed anisotropic transforms	24
3.6	Polyphase representation of a 1-D filter-bank	27
3.7	Equivalent polyphase representation of the lattice-based filtering and subsampling operations	28
4.1	Example of the construction of the multi-directional wavelet bases	33
4.2	Cyclic construction and basis functions	35
4.3	Unbalanced construction and basis functions	38
4.4	Two-directional basis functions	39
5.1	An example of NLA of a C^2/C^2 image	45
5.2	An example of NLA of the natural image <i>Cameraman</i>	47
5.3	Adaptation of transform directions in spatial segments	48
5.4	A typical convex hull of the R-D curve	50
5.5	A typical transform coder	52
5.6	Standard zerotrees of wavelet coefficients	53
5.7	Scanning order in the encoding process	54
5.8	The flow chart of encoding the significance map	54
5.9	Skewed anisotropic zerotrees defined in terms of lattices	57
5.10	Skewed anisotropic zerotrees across subbands	58
5.11	Full spatial quad-tree	60
5.12	Pruned spatial quad-tree	61
5.13	Circle and numerical results of compression	66
5.14	Optimal segmentation and choice of transform directions for circle	67
5.15	Reconstructions of circle	67
5.16	Details of reconstruction of circle	68
5.17	An image from the class C^2/C^2 and numerical results of compression	69

5.18	Optimal segmentation and choice of transform directions for C^2/C^2 image	70
5.19	Reconstructions of C^2/C^2 image	70
5.20	Lena and compression results	71
5.21	Barbara and compression results	71
5.22	Goldhill and compression results	72
5.23	Compression of Lena at low bitrate	73
5.24	E and S-type coefficients in a C^2/C^2 image	76
5.25	Anisotropic segmentation	77
6.1	Neighborhood of transform coefficients at two consequent scales	83
6.2	Isotropic and anisotropic neighborhood	89
6.3	Smooth denoising of a toy example	91
6.4	Oriented denoising of a toy example	93
6.5	Magnitudes produced by HP filtering along 4 directions	94
6.6	Energy of HP coefficients across directions for edge and smooth regions	95
6.7	Coefficient α in linear combination of denoising outputs	97
6.8	Numerical results of denoising of Barbara	99
6.9	Denoised versions of Barbara	100

List of Tables

2.1	Orders of approximation by the anisotropic transforms	15
3.1	Orders of approximation by the skewed anisotropic transforms	26
5.1	Dependence of the approximation rate on the growth rate of the number of transform directions	46
6.1	Denoising results	101

List of Algorithms

5.1	Bisection algorithm	51
5.2	Standard zerotree-based encoding algorithm	56
5.3	Optimized anisotropic zerotree-based encoding algorithm with spatial segmentation	63
6.1	GSM-based denoising algorithm	87
6.2	Combined denoising algorithm	98

Chapter 1

Introduction

1.1 Motivation

The problem of finding efficient representations of images is a fundamental problem in many image processing areas, including denoising, compression and feature extraction. An efficient transform-based representation requires sparsity, that is, a large amount of information has to be contained in a small portion of transform coefficients.

Geometrical features in images, like edges and contours, play one of the most important roles in the human visual system, since they carry most of the perceptual information. This fact is well known from the image processing [48] and computer vision [3] literature. These features are discontinuities in the two-dimensional (2-D) image domain, but they exhibit some degree of smoothness along a one-dimensional (1-D) curve. An efficient representation has to be capable of precise modeling and of providing a sparse description of this geometrical information.

The 1-D wavelet transform (WT) has become very successful in the last decade because it provides an efficient multiresolution representation of 1-D piecewise smooth signals [35; 101]. The application of wavelets to image processing requires the design of two-dimensional (2-D) wavelet bases. The most common approach is to construct such bases using 2-D separable filter-banks, which consist of the direct product of two independent 1-D filter-banks in the horizontal and vertical directions. Filtering using high-pass (HP) filters with enough vanishing moments (or zeroes at the corresponding frequency response at $\omega = 0$) along these two directions leads to a sparse representation of smooth signals. This method is conceptually simple and has very low complexity while all the 1-D wavelet theory carries over. These are the main reasons why it has been adopted in the image compression standard JPEG-2000 [86; 91].

Some other notable approaches use non-separable filter-banks and subsampling (e.g. quincunx) [17; 51; 52], but these methods are computationally complex and the design of the associated 2-D filter-banks is often challenging and much more involved. Also, several general multi-dimensional multi-channel filter design methods have been proposed in [4; 16; 67; 92] resulting in filters with separable polyphase components. In this thesis, we focus on the design and applications of 2-D separable two-channel filter-banks based on the 1-D wavelets, but allowing directionality and anisotropy.

Despite their success, the standard separable 2-D WT fails to provide a sparse represen-

tation in the presence of 1-D discontinuities, like edges or contours. These discontinuities, being highly anisotropic objects present in images, are characterized by a geometrical coherence that is not properly captured by the standard isotropic WT. Namely, many wavelets intersect a discontinuity and this leads to many large magnitude coefficients.

The reason for the inefficiency of the standard 2-D WT resides in the *spatial isotropy* of its construction, that is, filtering and subsampling operations are applied equally along the horizontal and vertical directions at each scale. As a result, the corresponding filters, obtained as the direct products of the 1-D filters, are isotropic at all scales.

This motivates us to design *anisotropic basis functions* that can “match” anisotropic objects. However, ensuring an efficient matching between anisotropic basis functions and objects in images is a non-trivial task. Anisotropic basis functions have already been considered and exploited by adaptive (e.g. bandelets [64; 65; 66]) or non-adaptive (edgelets and wedgelets [31; 74; 75; 107; 108], curvelets [9; 10; 11], contourlets [27; 28], etc.) processing. These methods build dictionaries of anisotropic basis functions that provide a sparse representation of edges in images. Furthermore, Candès and Donoho [10] showed that the parabolic scaling relation between the length and width of basis functions is a key feature to achieve a good non-linear approximation (NLA) behavior. However, the implementation of these transforms usually requires *oversampling*, having *higher complexity* when compared to the standard WT, and require *non-separable processing* (convolution) and *non-separable filter design*. Furthermore, in some of these constructions (e.g. curvelets [10]) the design of the associated filters is performed in the *continuous domain* and this makes it difficult to use them directly on discrete images and to achieve perfect reconstruction.

Notice that the standard 2-D WT uses only horizontal and vertical directions and the HP filters in this transform have vanishing moments only along these directions. Since characterization of features in synthetic and natural images involves many more than these two standard directions, *multi-directionality* (M-DIR) and *directional vanishing moments* (DVM) play an important role in pursuing sparse representations.

Several other approaches also analyze geometrical structures in images, like polynomial modeling with quadtree segmentation [83; 84], footprints and edgeprints [36; 37; 38], edge adapted multiscale transform [19], etc. Apart from the goal of efficient representation exploiting geometrical coherence, M-DIR processing has also been applied to image denoising and classification. Examples of such transforms are the steerable pyramids [85; 93], the cortex transform [109], the complex wavelets [50], the directional wavelet analysis [116], the directional filter-banks [4; 67; 77], brushlets [57], and the associative representation of visual information [41]. Some other methods involve directionally adaptive processing in order to preserve edges in images [44; 55; 60; 61], whereas the methods proposed in [23; 24] impose DVM in either critically sampled or oversampled filter-banks. However, all of them fail to provide a *perfect reconstruction* and *critical* and *separable sampling* while keeping filter design completely in the *discrete domain* and with filters having DVM along *arbitrary directions*.

Our goal is to construct an anisotropic perfect reconstruction transform with HP filters having DVM, while retaining the simplicity of 1-D processing and filter design from the standard separable 2-D WT. We propose a transform construction based on partitioning of the discrete space using integer lattices, where the 1-D filtering is performed along lines across the lattice. The corresponding basis functions are called *directionlets*. We show in this thesis that the critically sampled directionlets provide a successful representation for NLA and compression of synthetic and natural images, whereas the undecimated version of

directionlets is convenient for image denoising. We propose our construction for the 2-D case but it is realizable for any dimension.

The problem of NLA is defined as follows: an image (or signal, in general) is transformed and the N largest magnitude coefficients are retained, while the rest is set to zero. The reconstructed approximation of the original image is obtained as the inverse transform of the thresholded transform coefficients. The numerical result of NLA is expressed in terms of the *mean-square error* (MSE) defined as

$$\text{MSE} = \frac{1}{N_x N_y} \|\mathbf{x} - \hat{\mathbf{x}}\|^2 = \frac{1}{N_x N_y} \sum_{n=1}^{N_x N_y} (x_n - \hat{x}_n)^2, \quad (1.1)$$

where $\hat{\mathbf{x}}$ is an approximation of the original image \mathbf{x} and N_x and N_y are the dimensions of the image. For orthogonal transforms, the norm $\|\mathbf{x} - \hat{\mathbf{x}}\|^2$ that is related to the original image domain is equal to the sum of squares of the discarded transform coefficients y_n , that is,

$$\|\mathbf{x} - \hat{\mathbf{x}}\|^2 = \sum_{n=N+1}^{N_x N_y} y_n^2. \quad (1.2)$$

Recall from [56] that, in the case of orthogonal transforms, the optimal NLA approach in order to minimize the MSE given by (1.2) is to sort the transform coefficients y_n in the descending order of magnitudes. If the representation is complete, then the MSE decays, as N grows. The asymptotic rate of decay of the MSE, as $N \rightarrow \infty$, is the most important approximation property of a transform in terms of NLA. Here, we show that directionlets applied in NLA outperform the standard 2-D WT and achieve an asymptotic rate of decay for the MSE that is competitive with the rates achieved by other state-of-the-art *oversampled* methods.

Compression using orthogonal transforms is an extension of NLA involving indexing and quantization of the retained transform coefficients. The reconstructed images are compared in terms of an operational *rate-distortion* (R-D) performance, where, given R encoding bits, the distortion D is equal to the MSE of the reconstruction. Notice that the distortion is conserved in the case of *orthogonal transforms* making it easier to apply the Lagrangian based optimization methods. The application of directionlets in the wavelet-based compression methods [79; 81; 113; 114] is straightforward substituting the standard 2-D WT by the oriented anisotropic transforms. We show that an adaptive algorithm based on directionlets and allowing for spatial segmentation achieves compression results that outperform the results obtained by the standard 2-D WT.

Furthermore, the undecimated version of directionlets allows for an efficient denoising algorithm, where the original image is estimated from a noisy observation. The coherence among transform coefficients across space, directions and scales is efficiently enforced by the M-DIR multiresolution decomposition performed by directionlets. The obtained results are comparable to the current best results [23; 68; 87], while the computational complexity of directionlets is lower than the complexity of other denoising methods based on *non-separable transforms*.

Therefore, to summarize, we aim at the following goals in the thesis:

1. To propose a transform construction that inherits the separability, the computational simplicity and the simplicity of filter design from the standard separable 2-D WT, but that can improve efficiency of the representation of discontinuity curves in images.

2. To design practical methods that can provide a better approximation and compression power (that is, a faster asymptotic decay of the mean-square error and better rate-distortion performance) for synthetic and natural images, while keeping complexity low.
3. To come up with a denoising method based on the undecimated version of the transform, which can efficiently exploit correlation among wavelet coefficients across space, scales, and angles.

1.2 Related Work

The 2-D transforms that involve anisotropic basis functions and M-DIR processing have been topics of thorough research in the recent literature concerning their construction, properties, and the NLA behavior on piecewise smooth images. Example of these transforms are (as already mentioned in Section 1.1) bandelets [64; 65; 66], edgelets and wedgelets [31; 74; 75; 107; 108], curvelets [9; 10; 11], and contourlets [27; 28]. On the one hand, bandelets and wedgelets provide the optimal asymptotic decay of the MSE of approximation of C^2/C^2 images,¹ that is, $\text{MSE}=O(N^{-2})$. On the other hand, curvelets and contourlets perform nearly optimally, achieving an asymptotic decay given by $\text{MSE}=O(N^{-2} \log^3 N)$. However, notice that these transforms have the disadvantage that they are overcomplete and require non-separable processing.

Several other approaches also exploit geometrical coherence in images. Shukla et al. have applied piecewise polynomial modeling on spatial segments of images and compute the optimal quadtree spatial segmentation in order to minimize the distortion for a given bitrate [84]. Dragotti et al. have proposed edgeprints for image representation as an extension of the footprints to 2-D joining the footprint coefficients generated by smooth discontinuities in images [37]. Cohen and Matei have studied non-separable multi-scale transforms adapted to edges in images [19].

M-DIR processing is applied in several other methods. For instance, an example of such methods is given in [4; 67; 77], where the oriented filter-banks are designed in different directions. Furthermore, directionality in [50] is obtained using complex wavelets, whereas the construction proposed in [116] uses directional wavelets. Steerable filter-banks that are designed to have directional selectivity in the Fourier domain are given in [85]. Brushlets [57] also provide an efficient M-DIR description of images, whereas the methods in [41; 109] handle with directionality in visual representation in a more general way.

Compression of images has been a hot topic in the image processing literature. Certainly, the most popular result is the wavelet-based image coding standard JPEG-2000 [86; 91]. However, wavelet-based image coders have been proposed well before. Thus, scalar quantization and optimal bit allocation have been proposed in [2]. An efficient coding method that uses vector quantization of coefficients based on zerotree structures of wavelets has been presented in [81], followed by the even more efficient SPIHT algorithm [79]. The coding by spatial-frequency quantization [113; 114] optimizes the decomposition of the image in space and frequency using the Lagrangian optimization method. This coding algorithm uses quad-tree spatial segmentation, which is the most common approach in tiling-based image

¹Piecewise twice continuously differentiable images with discontinuities that are 1-D twice continuously differentiable functions.

coding methods because of its efficient coding. The quad-tree spatial segmentation has been introduced in image compression in [53] and also used later in [84; 89; 90; 96; 112]. On the other hand, more complex spatial segmentation methods have been analyzed in [45; 46; 47].

Denosing of images has been also a topic of interest in the recent literature. The wavelet-based thresholding methods have been introduced in [20; 30; 32; 33; 34]. Some alternative approaches can be found in [12; 14; 15; 22; 49; 78]. Several M-DIR transforms have been successfully applied in denoising methods. As an example, in [68], the Gaussian scale mixture model is used for estimation of the transform coefficients obtained by the steerable pyramids from a noisy observation. Furthermore, thresholding of the transform coefficients is exploited in [87] for curvelets and in [23] for contourlets to obtain denoised version of images.

1.3 Thesis Outline and Contribution

The main goal of the thesis is to design new M-DIR anisotropic transforms that retain separability and simplicity of the standard 2-D WT, while being capable of capturing geometrical features in images.

In Chapter 2, we propose new transform constructions that lead to anisotropic basis functions. These basis functions are more convenient to match anisotropic features in images, like edges and contours. We show that anisotropy contributes to sparsity in the description of these objects.

Chapter 3 presents the construction of skewed transforms based on integer lattices. These transforms allow to iterate 1-D filters efficiently across different directions. The corresponding basis functions have directional vanishing moments along two directions different from horizontal and vertical. These basis functions are called directionlets. The polyphase representation associated to the skewed transforms allows for an efficient implementation since the polyphase matrix is block-diagonal.

The construction of general M-DIR transforms is described in Chapter 4. Directional vanishing moments are imposed along a set of arbitrary directions in the M-DIR basis functions, while critical sampling is retained. However, we illustrate experimentally that it is difficult to have regularity of the basis functions in the general construction.

Chapter 5 studies the non-linear approximation power of directionlets using spatial segmentation of the image that reduces the number of dominant directions in each segment. The critically sampled directionlets outperform the standard 2-D WT in asymptotic approximation of piecewise smooth images and are comparable to other oversampled methods. We also apply directionlets to the field of image compression, where directionlets substitute the standard WT in the wavelet-based image coders, while spatial segmentation ensures that the orientation of directionlets is matched with the local dominant directions across the image domain. The numerical results of compression and the visual quality of the reconstructed images obtained by directionlets are substantially improved compared to the case of the standard WT.

In Chapter 6, we propose an image denoising method based on the undecimated version of directionlets. The transform coefficients are estimated from a noisy observation using the Gaussian scale mixtures. The results obtained by the method are comparable to the current state-of-the-art denoising results, while having lower complexity.

Finally, we conclude and summarize the key results in Chapter 7, where we also discuss

future research work.

Chapter 2

Anisotropic Two-Dimensional Wavelet Decompositions

2.1 Introduction

As explained in Chapter 1, the standard two-dimensional (2-D) wavelet transform (WT) fails to provide a compact representation in the presence of one-dimensional (1-D) discontinuities, like edges or contours. Many wavelets that intersect the discontinuity curve have large magnitudes.

The main reason for this inefficiency is the *spatial isotropy* of the construction of the standard 2-D WT, that is, filtering and subsampling operations are applied equally along both the horizontal and vertical directions at each scale. The corresponding filters, obtained as direct products of the 1-D counterparts, are isotropic at all scales (see Section 2.2).

However, discontinuity curves are highly anisotropic features present in images and they are characterized by a geometrical coherence that is not properly captured by the isotropic transforms. Such anisotropic objects generate many large magnitude wavelet coefficients, as shown in a simple example in Figure 2.1(a).

Our goal is to design *anisotropic basis functions* that can “match” anisotropic objects (Figure 2.1(b)). However, the design of the anisotropic transforms that ensures an efficient matching between the basis functions and objects in images is challenging.

In some recent work, anisotropic basis functions have already been proposed using adaptive (e.g. bandelets [64; 65; 66]) and non-adaptive (edgelets and wedgelets [31; 74; 75; 107; 108], curvelets [9; 10; 11], contourlets [27; 28], etc.) constructions. These methods build dictionaries of anisotropic basis functions that provide a sparse representation of edges in images. Furthermore, Candès and Donoho [10] showed that the necessary condition to have a good non-linear approximation (NLA) performance is that the length and width of basis functions obey the parabolic scaling relation. However, the implemented transforms are often *oversampled* and require *non-separable processing* (convolution) and *non-separable filter design*. Moreover, in some of these constructions (like curvelets) the design of the associated filters is performed in the *continuous domain* and this makes it difficult to use them directly on discrete images.

Here, we propose a new modified method that retains the 1-D filtering and subsampling

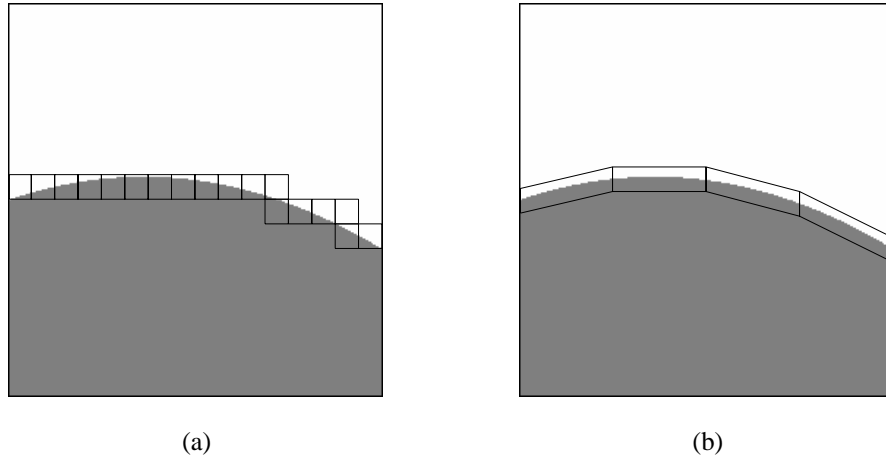


Figure 2.1: A simple image with one discontinuity along a smooth curve is represented by the two types of basis functions: isotropic and anisotropic. The support of these basis functions is shown schematically as black rectangles. (a) Isotropic basis functions generate a large number of significant coefficients around the discontinuity. (b) Anisotropic basis functions trace the discontinuity line and produce just a few significant coefficients.

operations and can provide anisotropy. In the sequel of this chapter, we first review the standard 2-D WT construction and, then, we give two examples of constructions of the anisotropic transforms that still inherit the simplicity of processing and filter design from the standard WT. Furthermore, these two anisotropic transforms are critically sampled and lead to perfect reconstruction.

2.2 Standard Decomposition

The most common design of the 2-D wavelet filter-banks is separable, where the associated basis functions are obtained as the direct product of two independent 1-D basis functions in the horizontal and vertical directions. Filtering with high-pass (HP) filters with enough vanishing moments along these two directions results in a sparse representation of smooth signals [101]. This method is conceptually simple and has very low computational complexity.

The construction of the standard 2-D WT is shown in Figure 2.2. The filtering and sub-sampling operations in the transform are iterated with an equal number of steps along both the horizontal and vertical directions at each scale. The basic 1-D low-pass (LP) and HP filters used in the transform are denoted as $H_0(z)$ and $H_1(z)$, respectively. The resulting filters, obtained as direct products of the 1-D counterparts, are isotropic at all scales.

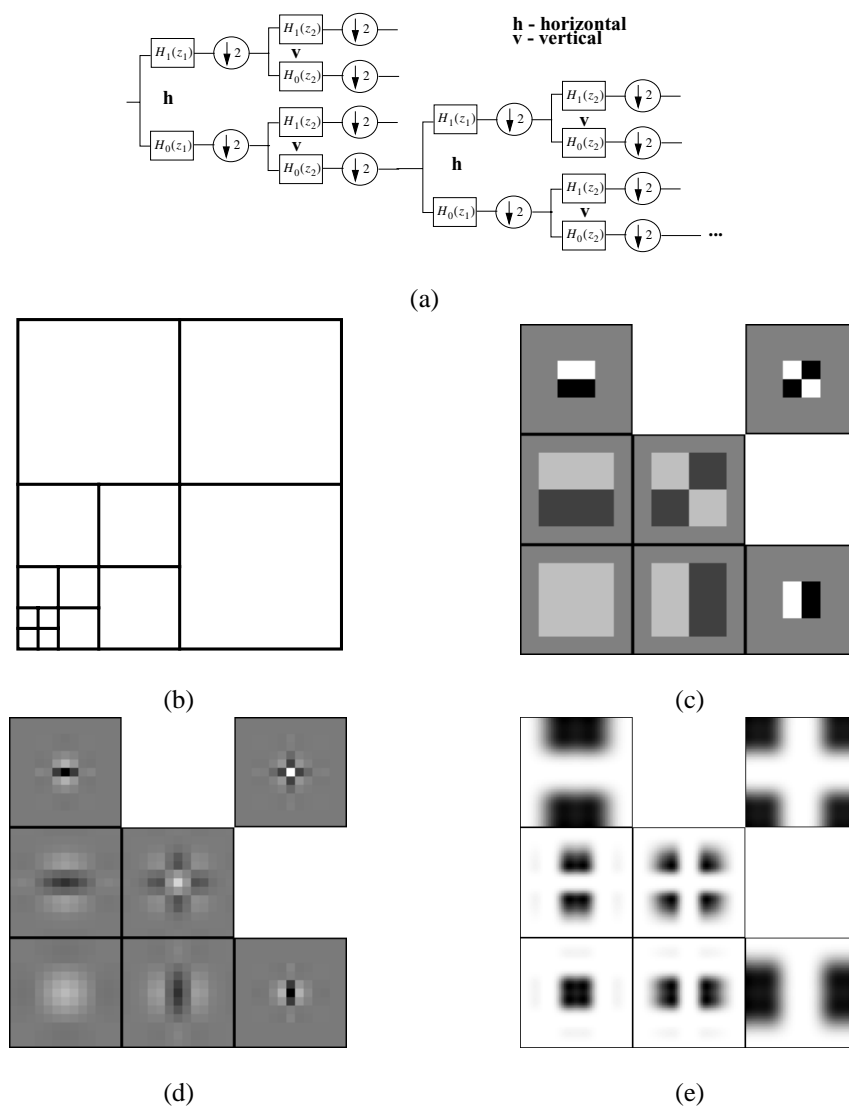


Figure 2.2: The standard 2-D WT is isotropic. (a) The filtering and subsampling operations are applied equally in both directions at each scale of the transform. (b) The corresponding decomposition in frequency. The basis functions obtained in this way are isotropic at each scale as shown in (c) for Haar and in (d) for biorthogonal "9-7" 1-D scaling and wavelet functions. (e) The corresponding Fourier transforms of the basis functions obtained from the "9-7" 1-D filters.

2.3 Fully separable decomposition

Define a simple class of piecewise polynomial images, denoted as $\text{Mondrian}(k_1, k_2)$ and inspired by the *geometrical period* of Piet Mondrian¹ [58].

Definition 2.1 *The class $\text{Mondrian}(k_1, k_2)$ contains $M \times M$ piecewise polynomial images with k_1 horizontal and k_2 vertical discontinuities.*

An example of the image from the class $\text{Mondrian}(k_1, k_2)$ is shown in Fig. 2.3(a). This class is not efficiently represented by the standard WT. The discontinuities lead to too many nonzero coefficients, as shown in the lemma below and in Fig. 2.3(b).

Lemma 2.1 *Given an $M \times M$ pixel image from the class $\text{Mondrian}(k_1, k_2)$, the number of nonzero transform coefficients in band-pass subbands produced by the standard WT with the 1-D wavelets having enough vanishing moments² is given by*

$$N = O((k_1 + k_2)M). \quad (2.1)$$

Proof: The three band-pass subbands at the j th ($1 \leq j \leq \log_2 M$) level of the standard WT contain $O(k_1 M/2^j + k_2)$, $O(k_1 + k_2 M/2^j)$, and $O(k_1 + k_2)$ nonzero coefficients. The total number of nonzero coefficients across scales is given by

$$\begin{aligned} N &= \sum_{j=1}^{\log_2 M} \left(O\left(k_1 \frac{M}{2^j} + k_2\right) + O\left(k_1 + k_2 \frac{M}{2^j}\right) + O(k_1 + k_2) \right) \\ &= O(2(k_1 + k_2) \log_2 M) + O((k_1 + k_2)(M - 1)) = O((k_1 + k_2)M). \end{aligned}$$

□

To improve compactness of the representation of the class $\text{Mondrian}(k_1, k_2)$, we define the *fully separable WT* (FSWT). In this transform a full 1-D WT is applied in the horizontal direction (each row of image) and then, on each output a full 1-D WT is applied in the vertical direction (each column). The decomposition scheme is shown in Figure 2.4(a). Notice that such a decomposition has already been proposed in [110] and also in [62; 76], where it is referred to as tensor wavelet basis.

The FSWT provides anisotropic basis functions (Figure 2.4(c)) that are better adapted to the anisotropic objects such as the discontinuities in the class $\text{Mondrian}(k_1, k_2)$. Representation efficiency is strongly improved, as can be seen in Figure 2.3(c) from the resulting sparsity and it is given in Lemma 2.2.

Lemma 2.2 *Given an $M \times M$ pixel image from the class $\text{Mondrian}(k_1, k_2)$, the number of nonzero transform coefficients in band-pass subbands produced by the FSWT with the 1-D wavelets having enough vanishing moments is given by*

$$O\left((k_1 + k_2)(\log_2 M)^2\right). \quad (2.2)$$

¹The Dutch painter established *neoplasticism* and *De Stijl* in Europe in the beginning of the 20th century. The image shown in Fig. 2.3(a) resembles to the paintings from his *geometrical period* (1930)

²A polynomial of the n th order is annihilated by a wavelet that has at least $n + 1$ vanishing moments.

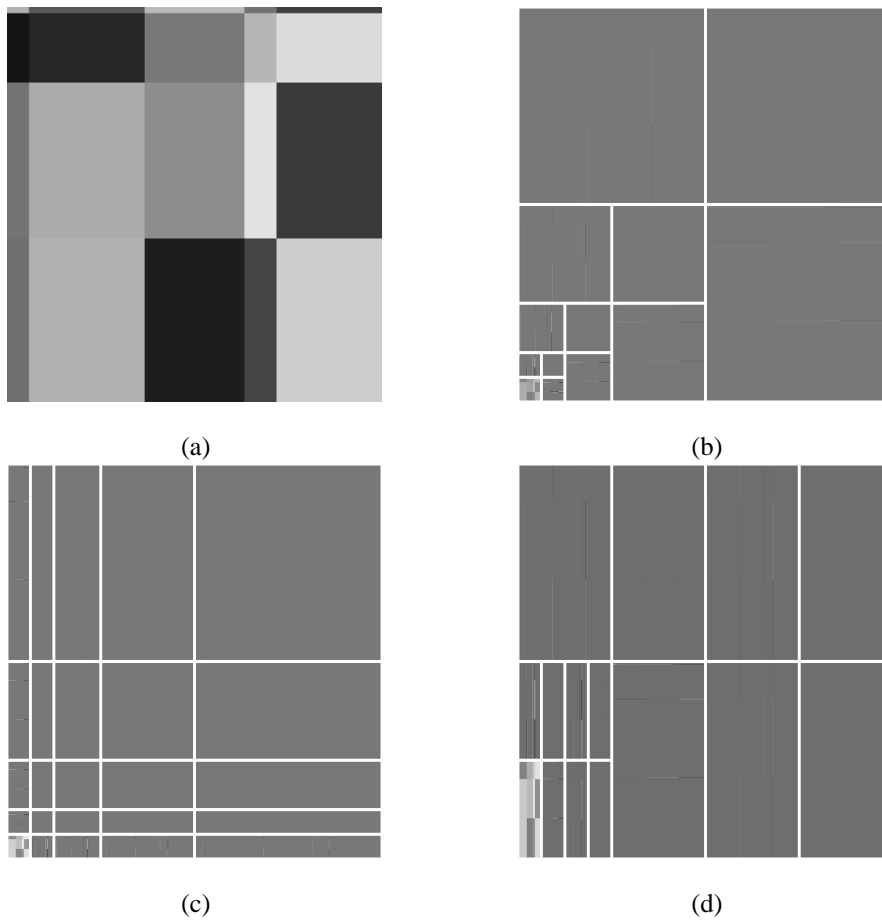


Figure 2.3: The transforms of an image from the class $\text{Mondrian}(k_1, k_2)$. (a) An image from the class $\text{Mondrian}(k_1, k_2)$. This class is inspired by the painting style established by Piet Mondrian (1872 - 1944). The image is transformed by the three transforms: (b) standard WT, (c) FSWT, (d) AWT(2,1) with 1-D wavelet filters having enough vanishing moments.

Proof: Each band-pass subband is indexed by (j_1, j_2) , where j_1 determines the number of the horizontal transforms, whereas j_2 enumerates the vertical transforms. The indexes are in the range $1 \leq j_1, j_2 \leq \log_2 M$.

The subband (j_1, j_2) contains $O(k_1 + k_2)$ nonzero transform coefficients, therefore, the total number of nonzero coefficients is given by

$$N = \sum_{j_1=1}^{\log_2 M} \sum_{j_2=1}^{\log_2 M} O(k_1 + k_2) = O\left((k_1 + k_2)(\log_2 M)^2\right).$$

□

The performance of the FSWT on the class $\text{Mondrian}(k_1, k_2)$, given by (2.2), is substantially better than the result of the standard WT, given by (2.1), namely, there is an exponential improvement in terms of M . The improvement is a consequence of anisotropy of the basis functions that is matched to the anisotropy of the class. However, the FSWT performs well only when it is applied on Mondrian-like images, while natural images contain features that are not well represented by straight (horizontal and vertical) lines.

Notice that if a transformed image contains a curve (or any discontinuity that is not a straight line), then the FSWT fails, as the number of nonzero coefficients grows exponentially across scales. Intuitively, the failure happens because the FSWT enforces a higher anisotropy (or elongation of the basis functions) than the one that is required in order to provide a compact representation of objects in natural images. To overcome this problem, we introduce a novel anisotropic transform, which performs better on a larger class of images.

2.4 Anisotropic wavelet decomposition

In the *anisotropic WT* (AWT) the number of transforms applied along the horizontal and vertical directions is unequal, that is, there are n_1 horizontal and n_2 vertical transforms at a scale, where n_1 is not necessarily equal to n_2 . Then, the iteration is continued in the low-pass (LP), like in the standard WT. We denote such an anisotropic transform as $\text{AWT}(n_1, n_2)$. The *anisotropy ratio* $\rho = n_1/n_2$ determines elongation of the basis functions of the $\text{AWT}(n_1, n_2)$. An example of the construction and basis functions is shown in Figure 2.5, where the $\text{AWT}(2, 1)$ is used.

Notice that both the standard WT and the FSWT can be expressed in terms of the AWT. The standard WT is simply given by $\text{AWT}(1, 1)$. However, the representation of the FSWT is more complex and is given as a concatenation of two AWTs. The first transform is $\text{AWT}(n_{1max}, 0)$ that produces $n_{1max} + 1$ subbands and it is followed by the $\text{AWT}(0, n_{2max})$ applied on each subband. The arguments n_{1max} and n_{2max} determine the maximal number of transforms in the two directions and depend on the size of the image.

Even though the AWT is not the most appropriate representation for the particular case of Mondrian-like images, it improves approximation of more general classes of images, as shown in Chapter 5. Figure 2.3(d) shows the result of the $\text{AWT}(2, 1)$ of an image from the class $\text{Mondrian}(k_1, k_2)$. The order of the number of nonzero coefficients is given by the following lemma.

Lemma 2.3 *Given an $M \times M$ pixel image from the class $\text{Mondrian}(k_1, k_2)$, the number of nonzero transform coefficients in band-pass subbands produced by the $\text{AWT}(n_1, n_2)$ with 1-D*

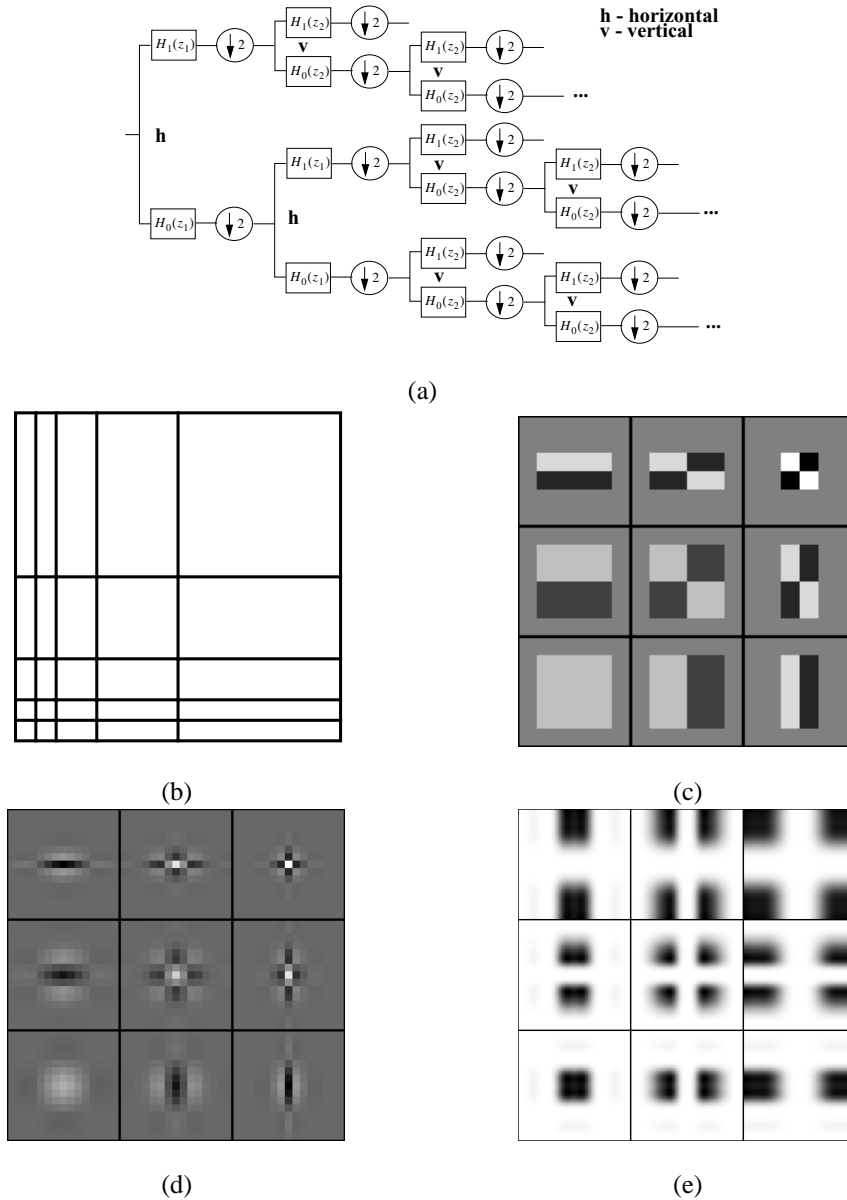


Figure 2.4: The FSWT is anisotropic, as the number of 1-D transforms is not equal in the two directions. (a) An example of the transform scheme. Only 2 steps in each direction are shown. (b) The decomposition in frequency that corresponds to the construction in (a) with 4 steps in each direction. The anisotropic basis functions obtained from the (c) Haar and (d) biorthogonal "9-7" 1-D scaling and wavelet functions. (e) The corresponding Fourier transform of the basis functions obtained from the "9-7" 1-D filters.

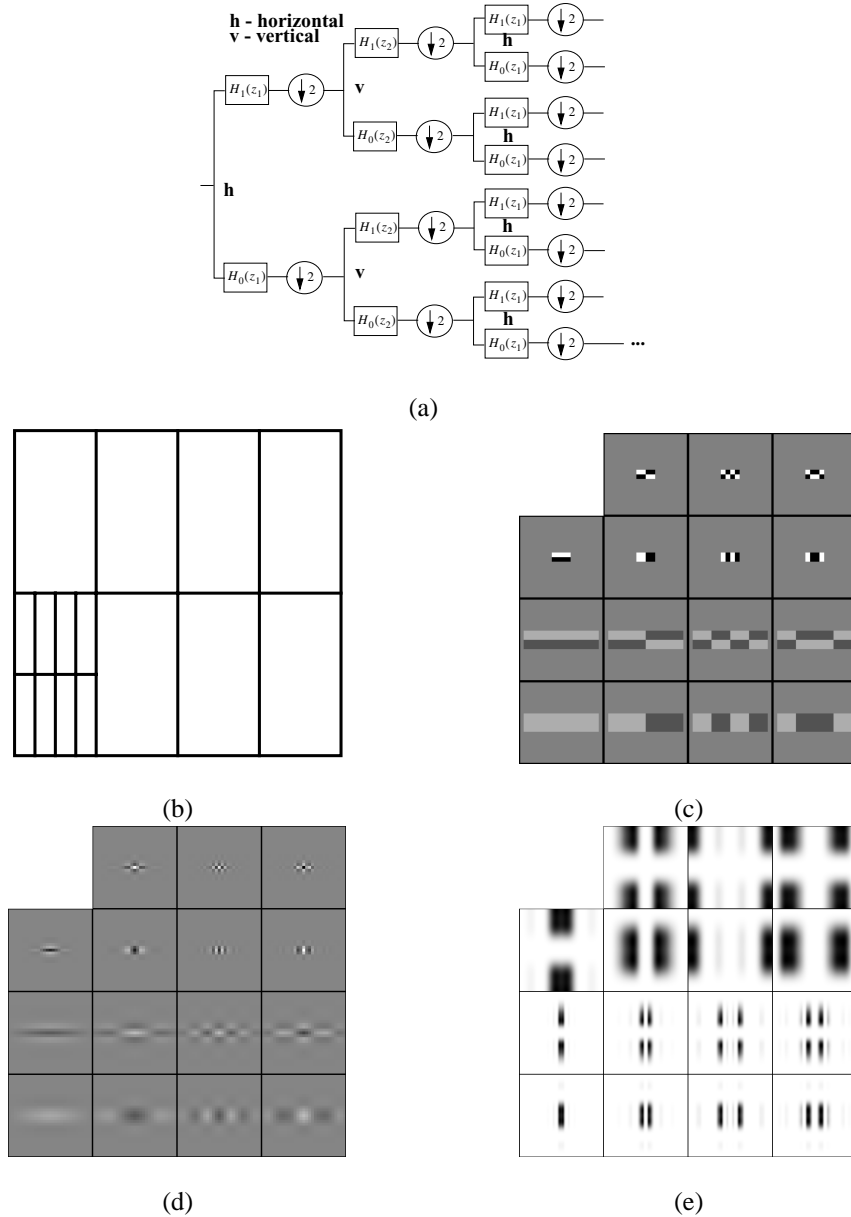


Figure 2.5: The AWT allows for anisotropic iteration of the filtering and subsampling applied on the LP, similarly as in the standard WT. Although this transform does not improve approximation of the class $\text{Mondrian}(k_1, k_2)$, it provides an efficient approximation tool for more general classes of images (Chapter 5). (a) The filtering scheme for the AWT(2,1), where one step of iteration is shown. (b) The decomposition in frequency. The basis functions obtained from the (c) Haar and (d) biorthogonal "9-7" 1-D scaling and wavelet functions. (e) The corresponding Fourier transform of the basis functions obtained from the "9-7" 1-D filters.

Table 2.1: Orders of approximation by the standard WT, FSWT and AWT applied on the class $\text{Mondrian}(k_1, k_2)$.

Standard WT	FSWT	AWT
$(k_1 + k_2)M$	$(k_1 + k_2)(\log_2 M)^2$	$(k_1 a + k_2/a)M$

wavelets having enough vanishing moments is given by

$$O\left(\left(ak_1 + \frac{1}{a}k_2\right)M\right), \text{ where } a = \frac{2^{n_2} - 1}{2^{n_1} - 1}. \quad (2.3)$$

Proof: The number of nonzero coefficients produced at the j th level of the $\text{AWT}(n_1, n_2)$ is given by

$$\begin{aligned} n(j) = & O(k_1 (2^{n_2} - 1) \frac{M}{2^{n_1 j}} + k_1 (2^{n_1} - 1) 2^{n_2} \\ & + k_2 (2^{n_1} - 1) \frac{M}{2^{n_2 j}} + k_2 (2^{n_2} - 1) 2^{n_1}). \end{aligned}$$

The total number of nonzero coefficients across scales is, therefore,

$$N = \sum_{j=1}^{\frac{\log_2 M}{\max(n_1, n_2)}} n(j) = O\left(\left(ak_1 + \frac{1}{a}k_2\right)M\right).$$

□

Notice that the result in Lemma 2.3 is a generalization of the result in Lemma 2.1. Table 2.1 summarizes the orders of numbers of nonzero coefficients in band-pass subbands produced by the three transforms applied on the class $\text{Mondrian}(k_1, k_2)$.

The transforms explained in this chapter are applied in the horizontal and vertical directions only. More general transforms can be obtained by imposing vanishing moments along different directions. These transforms provide an efficient representation of more general classes of images, involving more than only the two standard directions, as shown in the next chapter.

2.5 Conclusions

In this chapter, we have presented two anisotropic transform constructions, which are called fully separable WT and anisotropic WT. We have shown that these transforms retain the simplicity of separable filtering and subsampling from the standard 2-D WT and provide anisotropic basis functions that can capture more efficiently anisotropic features in images, like edges and contours.

Chapter 3

Lattice-based Skewed Discrete Wavelet Transforms

3.1 Introduction

Several transform constructions that lead to anisotropic basis functions have been presented in Chapter 2. However, all the constructions, including also the standard two-dimensional (2-D) wavelet transform (WT), use only the horizontal and vertical directions. Notice also that the high-pass (HP) filters in these transforms have vanishing moments only along these two directions. Since characterization of features in synthetic and natural images involves many more than these two standard directions, *multi-directionality* (M-DIR) and *directional vanishing moments* (DVM) are very important in achieving sparse representation.

Multi-directionality has also been exploited in bandelets [64; 65; 66], edgelets and wedgelets [31; 74; 75; 107; 108], curvelets [9; 10; 11], and contourlets [27; 28]. Apart from the geometrical representation, several methods involve also multi-directional (M-DIR) processing, like the steerable pyramids [85], the cortex transform [109], the complex wavelets [50], the directional wavelet analysis [116], the directional filter-banks [4; 77], brushlets [57], and the associative representation of visual information [41]. However, all of them fail to provide a *perfect reconstruction* and *critically sampled separable* scheme with a filter design completely in the *discrete domain* and with filters having DVM along *arbitrary directions*.

Our goal in this chapter is to construct an anisotropic perfect reconstruction and critically sampled transform with HP filters having DVM along different directions, while retaining the simplicity of 1-D processing and filter design from the standard separable 2-D WT. Here, we present a novel M-DIR basis construction that is based on partitioning the discrete space using integer lattices.

In the continuation, we explain the problem of approximation of directions in the discrete space \mathbb{Z}^2 and we introduce the concept of directional interaction. Then, we propose the new lattice-based method that allows for a generalization of the anisotropic transform constructions from Chapter 2 to include separable (1-D) filtering and subsampling across multiple directions, not only horizontal and vertical. We also give the polyphase analysis of the lattice-based transforms.

3.2 Discretization of Directions

To apply a discrete transform in the discrete space \mathbb{Z}^2 in a certain direction, we need to define the pixels that approximate the chosen direction. This problem has been considered in computer graphics in the 1960's [7] as well as in [13; 40].

Recall that the set of points $(x, y) \in \mathbb{R}^2$ represents a continuous line with the slope r and intercept b if the following equality is satisfied:

$$y = rx + b. \quad (3.1)$$

The discrete approximation of (3.1) is called *digital line* $L(r, n)$ [100]. To preserve critical sampling in the transform, given a slope r , every pixel belongs to one and only one digital line $L(r, n)$. In that case, we say that, given a slope r , the set of digital lines $\{L(r, n) : n \in \mathbb{Z}\}$, partitions the discrete space \mathbb{Z}^2 .

The definitions of digital lines proposed in [7; 13; 40] are similar and here we give the definition that is a variation of the one given in [7]. We show also below that such digital lines partition the discrete space \mathbb{Z}^2 .

Definition 3.1 Given a rational slope r , the digital line $L(r, n)$, where $n \in \mathbb{Z}$, is defined as the set of pixels (i, j) such that

$$\begin{aligned} j &= \lceil ri \rceil + n, \quad \forall i \in \mathbb{Z}, \text{ for } |r| \leq 1, \text{ or} \\ i &= \lceil j/r \rceil + n, \quad \forall j \in \mathbb{Z}, \text{ for } |r| > 1. \end{aligned} \quad (3.2)$$

Lemma 3.1 Given a rational slope r , the set of digital lines $\{L(r, n) : n \in \mathbb{Z}\}$ partitions the discrete space \mathbb{Z}^2 .

Proof: We give the proof only for the case $|r| \leq 1$. Similar arguments can be used for the other cases.

For each pixel $(i, j) \in \mathbb{Z}^2$, we can find the intercept $n = j - \lceil ri \rceil$ such that the pixel belongs to the digital line $L(r, n)$. Furthermore, from (3.2) it follows that this intercept is unique. Therefore, the digital lines $L(r, n)$, $\forall n \in \mathbb{Z}$, partitions the discrete space \mathbb{Z}^2 . \square

The concept of digital lines is useful for overcomplete M-DIR representation. However, in the sequel, we show why digital lines do not provide an efficient framework when transforms are applied in different directions and critical sampling is enforced.

3.3 Directional Interaction

To explain the problem of *directional interaction*, let us first generalize the class `Mondrian` allowing for more directions. The class `S-Mondrian` consists of the skewed Mondrian-like images along two directions with the rational slopes $r_1 = b_1/a_1$ and $r_2 = b_2/a_2$, where a_1 , a_2 , b_1 , and b_2 are integers. To simplify notation, the two slopes are jointly denoted by the matrix

$$\mathbf{M}(r_1, r_2) = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}.$$

Definition 3.2 The class `S-Mondrian` $(\mathbf{M}(r_1, r_2), k_1, k_2)$ contains $M \times M$ piecewise polynomial images with k_1 and k_2 discontinuities along the digital lines $L(r_1, n)$ and $L(r_2, n)$, respectively, where $n \in \mathbb{Z}$, $r_1 = b_1/a_1$, $r_2 = b_2/a_2$, and $a_1, a_2, b_1, b_2 \in \mathbb{Z}$.

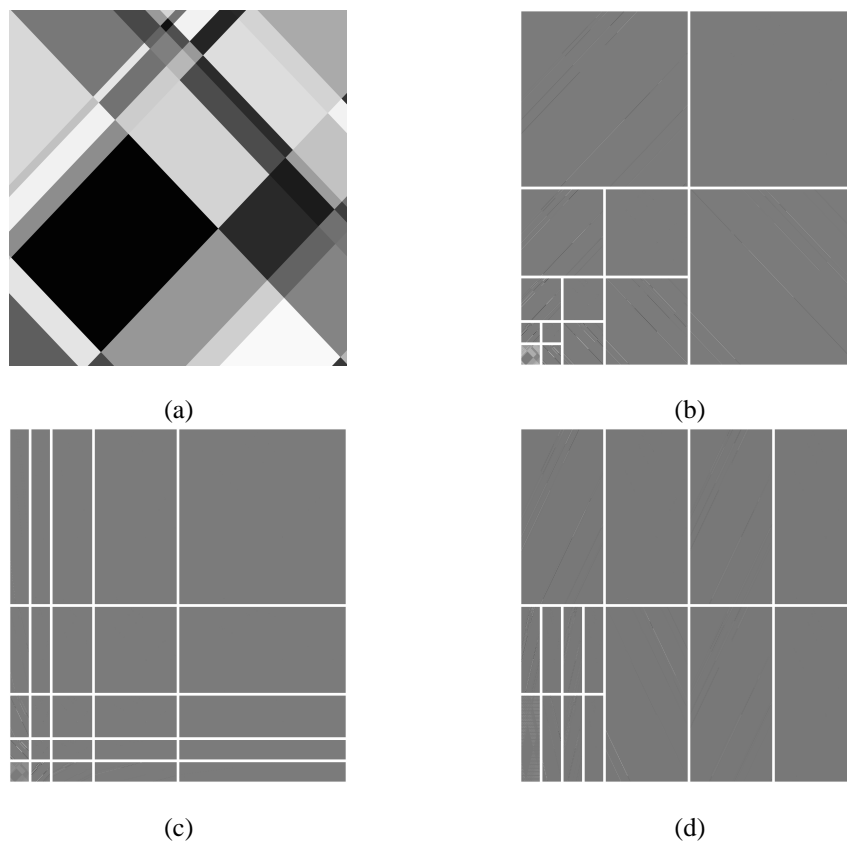


Figure 3.1: The transforms of an image from the class $S\text{-Mondrian}(\mathbf{M}(r_1, r_2), k_1, k_2)$ (a) An example of an image from the class $S\text{-Mondrian}(\mathbf{M}(r_1, r_2), k_1, k_2)$, for $\mathbf{M} = [\mathbf{v}_1, \mathbf{v}_2]^T$, where $\mathbf{v}_1 = [1, 1]$ and $\mathbf{v}_2 = [-1, 1]$. The image is transformed using (b) S-WT, (c) S-FSWT, and (d) S-AWT($\mathbf{M}_\Lambda, 2, 1$) (directionlets), where all the transforms are built on the lattice Λ determined by the generator matrix $\mathbf{M}_\Lambda = \mathbf{M}(r_1, r_2)$.

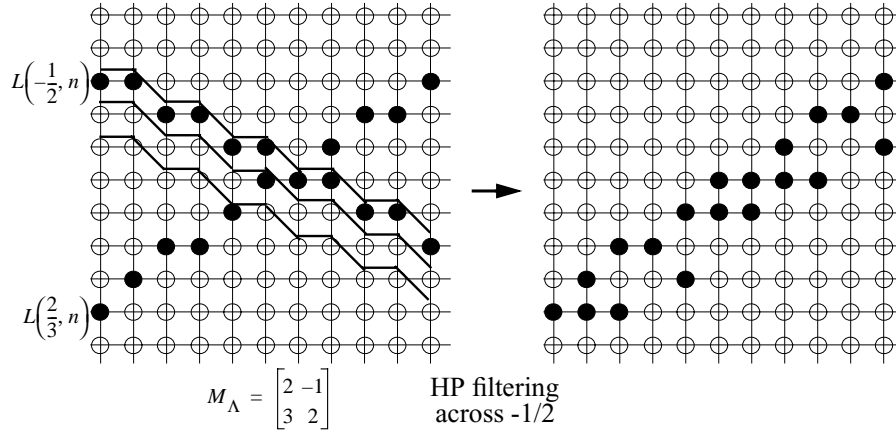


Figure 3.2: A 1-D WT is applied on an image from the class S-Mondrian $(M(-1/2, 2/3), 1, 1)$ along the digital lines $L(-1/2, n)$. The HP filtering annihilates the digital line with the slope $-1/2$. However, the nonzero coefficients produced by the other line with the slope $2/3$ are not aligned in the digital lines $L(2/3, n)$. This is called *directional interaction*. Although the transform along digital lines is efficient when applied in oversampled schemes, it fails to provide a systematic subsampling method when critical sampling is enforced.

Notice that the class Mondrian(k_1, k_2) is a special case of the larger class S-Mondrian $(M(r_1, r_2), k_1, k_2)$ when $M(r_1, r_2) = \mathbf{I}_2$. An example of an image from the class S-Mondrian $(M(r_1, r_2), k_1, k_2)$ is shown in Figure 3.1(a). Notice also that only the lines with rational slopes are used in the class S-Mondrian. However, in spite of this constraint, a wealth of directions is still available, as we will explain in Section 3.4.

To provide a sparse representation of the class S-Mondrian $(M(r_1, r_2), k_1, k_2)$ and following the ideas from Chapter 2, we apply a 1-D WT along the digital lines $L(r_1, n)$, for $n \in \mathbb{Z}$. The transform produces two types of nonzero coefficients, that is, the coefficients corresponding to the discontinuities with the slopes r_1 and r_2 .

Since the HP filter has vanishing moments along digital lines with the slope r_1 , the coefficients along this direction are annihilated in the HP subband, while the coefficients along the second direction with the slope r_2 are retained in both subbands. However, after subsampling, unlike in the case of the standard directions, the coefficients along the second direction are not aligned, that is, they cannot be clustered in the digital lines with the slope r_2 . Therefore, the following 1-D WT applied along the digital lines with the slope r_2 does not annihilate the coefficients along the second direction and, hence, it yields a non-sparse representation. We call this phenomenon *directional interaction*. The proof is trivial and is omitted here. An example is shown in Figure 3.2.

Notice also that the concept of digital lines does not provide a systematic rule for subsampling in the case of iteration of the filtering and subsampling along the directions with the slopes r_1 and r_2 when critical sampling is enforced. To overcome the directional interaction and to propose an organized iterated subsampling method we use the concept of integer lattices.

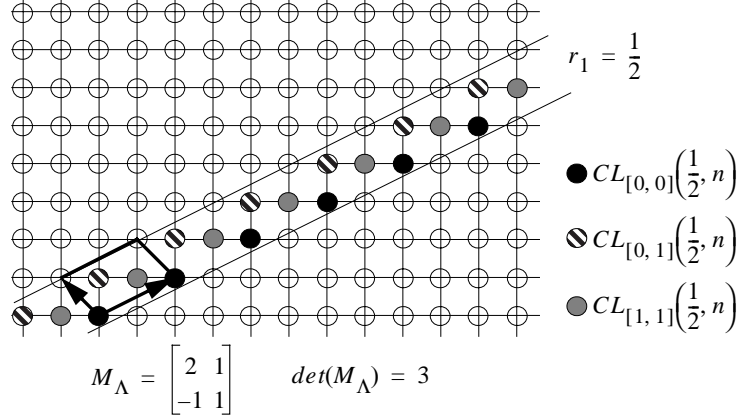


Figure 3.3: The intersections between the 3 cosets of the lattice Λ given by the generator matrix \mathbf{M}_{Λ} and the digital lines $L(r_1 = 1/2, n)$, where $n \in \mathbb{Z}$, are the co-lines $CL_{[0,0]}(1/2, n)$, $CL_{[0,1]}(1/2, n)$, and $CL_{[1,1]}(1/2, n)$.

3.4 Lattice-based Filtering and Subsampling

Instead of applying a transform along digital lines, we propose a novel method that is based on integer lattices [21]. We also prove that the lattice-based transforms can avoid directional interaction and are capable of providing the same order of approximation for the class S-Mondrian as the fully separable WT (FSWT) achieves for the class Mondrian.

A full-rank integer lattice Λ consists of the points obtained as linear combinations of two linearly independent vectors, where both the components of the vectors and the coefficients are integers. Any integer lattice Λ is a sublattice of the cubic integer lattice \mathbb{Z}^2 , that is, $\Lambda \subset \mathbb{Z}^2$. The lattice Λ can be represented by a non-unique generator matrix

$$\mathbf{M}_{\Lambda} = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix} = \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix}, \text{ where } a_1, a_2, b_1, b_2 \in \mathbb{Z}. \quad (3.3)$$

Recall that the cubic lattice \mathbb{Z}^2 can be partitioned into $|\det(\mathbf{M}_{\Lambda})|$ cosets of the lattice Λ [21], where each coset is determined by the shift vector \mathbf{s}_k , for $k = 0, 1, \dots, |\det(\mathbf{M}_{\Lambda})| - 1$. Therefore, the lattice Λ with the corresponding generator matrix \mathbf{M}_{Λ} given by (3.3), partitions each digital line $L(r_1 = b_1/a_1, n)$ into *co-lines*. Notice that a co-line is simply the intersection between a coset and a digital line. Similarly, the digital line $L(r_2 = b_2/a_2, n)$ is also partitioned into the corresponding co-lines (Figure 3.3).

We denote as $CL_{\mathbf{s}_k}(r_1, n)$ the co-line obtained as the intersection between the k th coset of the lattice Λ and the digital line $L(r_1 = b_1/a_1, n)$. Notice that the co-line $CL_{\mathbf{s}_k}(r_1, n)$ consists of the pixels $\{c_1 \mathbf{d}_1 + c_2 \mathbf{d}_2 + \mathbf{s}_k : \forall c_1 \in \mathbb{Z}, \text{ fixed } c_2 \in \mathbb{Z}\}$, where $n = \lceil c_2(b_2 - r_1 a_2) + s_{k,2} - r_1 s_{k,1} \rceil$ and $\mathbf{s}_k = [s_{k,1}, s_{k,2}]$.

Now we apply the 1-D WT (including the 1-D both filtering and subsampling operations) along the co-lines $\{CL_{\mathbf{s}_k}(r_1, n) : n \in \mathbb{Z}, k = 0, 1, \dots, |\det(\mathbf{M}_{\Lambda})| - 1\}$ (see also [97; 98; 99]). Notice that both filtering and subsampling are applied in each of the cosets separately. Furthermore, each filtering operation is purely 1-D. After subsampling, the retained points

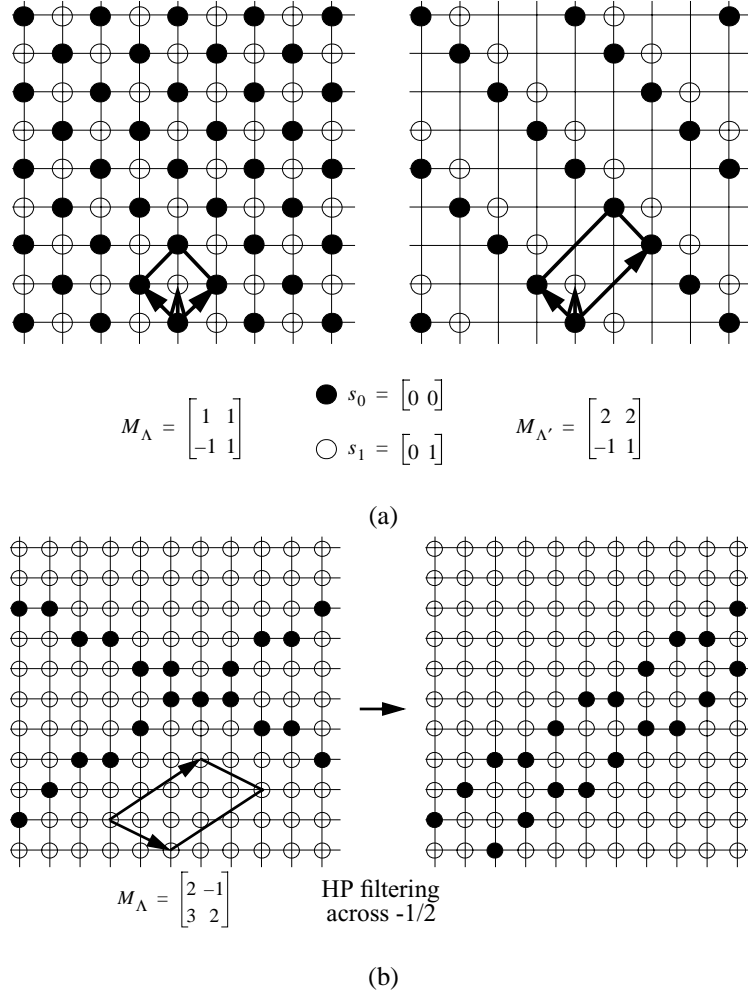


Figure 3.4: The lattice-based filtering and subsampling does not create directional interaction. (a) The lattice Λ is determined by the generator matrix M_{Λ} . 1-D Filtering is applied along the co-lines $\{CL_{s_k}(r_1, n) : n \in \mathbb{Z}, k = 0, 1, \dots, |\det(M_{\Lambda})| - 1\}$, where the slope r_1 corresponds to the vector $[1, 1]$, that is, along 45° . The pixels retained after the subsampling belong to the lattice $\Lambda' \subset \Lambda$ determined by the generator matrix $M_{\Lambda'}$. Notice that filtering and subsampling are applied separately in two cosets, determined by the shift vectors s_0 and s_1 . (b) The nonzero pixels obtained after one step of the lattice-based filtering operation applied on the same example as in Figure 3.2 are clustered in the digital lines with the slope $2/3$.

belong to the sublattice Λ' of the lattice Λ ($\Lambda' \subset \Lambda$) with the corresponding generator matrix given by (see Figure 3.4(a))

$$\mathbf{M}_{\Lambda'} = \mathbf{D}_s \cdot \mathbf{M}_{\Lambda} = \begin{bmatrix} 2\mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix}.$$

Here, \mathbf{D}_s is the horizontal subsampling operator, that is,

$$\mathbf{D}_s = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}.$$

We call the direction along the first vector \mathbf{d}_1 (with the slope $r_1 = b_1/a_1$), the *transform direction*. Similarly, the direction along the second vector \mathbf{d}_2 we call the *alignment direction*.

Therefore, since the filtering and subsampling are applied in each coset separately, the pixels retained after the subsampling are clustered in co-lines along the alignment direction. This property is crucial to avoid directional interaction (see Figure 3.4(b)).

Lemma 3.2 *Given a 1-D WT applied along the set of co-lines $\{CL_{\mathbf{s}_k}(r_1, n) : n \in \mathbb{Z}, k = 0, 1, \dots, |\det(\mathbf{M}_{\Lambda})| - 1\}$ on an image from the class $S\text{-Mondrian}(\mathbf{M}(r_1, r_2), k_1, k_2)$, the transform coefficients in band-pass subbands that correspond to the discontinuities with the slope r_2 are aligned, that is, they can be clustered in the co-lines $CL_{\mathbf{s}_k}(r_2, n)$, $n \in \mathbb{Z}$.*

Proof: Recall that the co-line $CL_{\mathbf{s}_k}(r_1, n)$ consists of the pixels $\{(i, j) : i = c_1 a_1 + c_2 a_2 + s_{k,1}, j = c_1 b_1 + c_2 b_2 + s_{k,2}, \forall c_1 \in \mathbb{Z}, \text{ fixed } c_2 \in \mathbb{Z}\}$. After the subsampling, the retained pixels belong to the lattice Λ' and, thus, the corresponding co-lines consist of the pixels (i, j) such that $i = c_1 \cdot 2a_1 + c_2 a_2 + s_{k,1}$ and $j = c_1 \cdot 2b_1 + c_2 b_2 + s_{k,2}$ for each $c_1 \in \mathbb{Z}$ and a fixed $c_2 \in \mathbb{Z}$.

Notice that the co-lines $CL_{\mathbf{s}_k}(r_2, n)$ with the other slope r_2 that correspond to the lattice Λ' consist of the same pixels. Therefore all the retained pixels are aligned in the direction with the slope r_2 . \square

Combining lattices with the different constructions given in Chapter 2, we build *skewed wavelet transforms*.

3.5 Skewed Wavelet Transforms

The transforms defined in Chapter 2 (the standard WT, FSWT, and anisotropic WT (AWT)) are inefficient when applied on the class $S\text{-Mondrian}(\mathbf{M}(r_1, r_2), k_1, k_2)$, unless $\mathbf{M}(r_1, r_2)$ is the identity matrix. Since the directions of the transforms and discontinuities in images are not matched, the transforms fail to provide a compact representation. The following lemma gives the orders of approximation that can be achieved by the three transforms with the standard directions.

Lemma 3.3 *Given an $M \times M$ pixel image from the class $S\text{-Mondrian}(\mathbf{M}(r_1, r_2), k_1, k_2)$, where $\mathbf{M}(r_1, r_2)$ is not the identity matrix, the standard WT, FSWT, and AWT with 1-D wavelets having enough vanishing moments provide $O((k_1 + k_2)M)$ nonzero transform coefficients in band-pass subbands.*

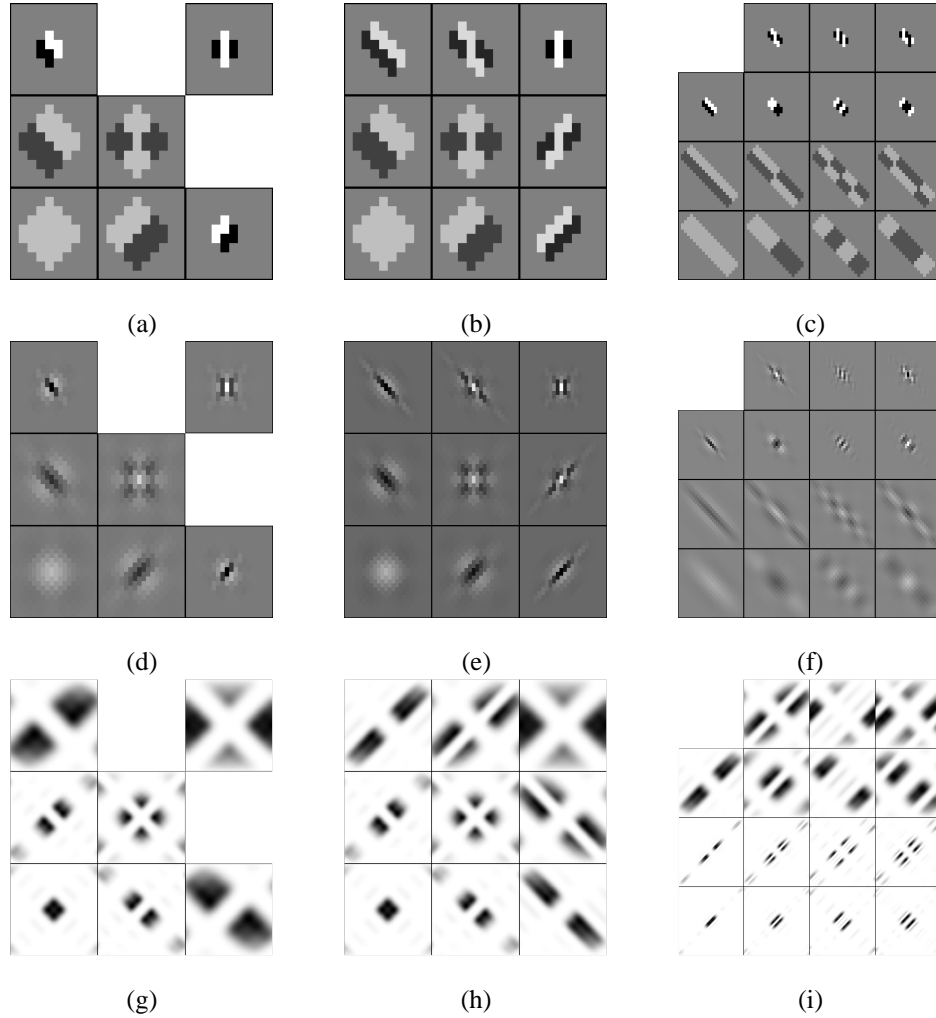


Figure 3.5: The basis functions obtained by the skewed transforms using the Haar 1-D scaling and wavelet functions: (a) S-WT, (b) S-FSWT, (c) S-AWT($M_\Lambda, 2, 1$) (directionlets). The same, but with the biorthogonal "9-7" 1-D scaling and wavelet functions: (d) S-WT, (e) S-FSWT, (f) S-AWT($M_\Lambda, 2, 1$) (directionlets). In all cases $M_\Lambda = [\mathbf{d}_1, \mathbf{d}_2]^T$, where $\mathbf{d}_1 = [1, 1]$, and $\mathbf{d}_2 = [-1, 1]$. The DVMs are imposed along the vectors \mathbf{d}_1 and \mathbf{d}_2 , that is, along 45° and -45° . The corresponding Fourier transforms: (g) S-WT, (h) S-FSWT, (i) S-AWT($M_\Lambda, 2, 1$) (directionlets).

Proof: The subbands produced by the FSWT are indexed by (j_1, j_2) , where $1 \leq j_1, j_2 \leq \log_2 M$. Each subband contains $O(k_1 M/2^{j_1} + k_2 M/2^{j_2})$ nonzero coefficients. The total number is given by

$$N = \sum_{j_1=1}^{\log_2 M} \sum_{j_2=1}^{\log_2 M} O\left(k_1 \frac{M}{2^{j_1}} + k_2 \frac{M}{2^{j_2}}\right) = O((k_1 + k_2)M).$$

Notice that the standard WT, as a special case of the AWT, has the same behavior. Thus, we give the proof only for the AWT. The $\text{AWT}(n_1, n_2)$ produces $2^{n_1+n_2} - 1$ band-pass and HP subbands at each scale j . Each of these subbands contain $n(j) = O((2^{n_1+n_2} - 1)M(2^{-n_1j} + 2^{-n_2j}))$ nonzero coefficients. Therefore, the total number of nonzero coefficients is given by

$$\sum_{j=1}^{\frac{\log_2 M}{\max(n_1, n_2)}} n(j) = O((k_1 + k_2)M).$$

□

Using integer lattices, we define the three new transforms, which are *skewed versions* of the standard WT, FSWT, and AWT. Given a lattice Λ , the skewed transforms are applied along co-lines in the transform and alignment directions of the lattice Λ , retaining the same frequency decompositions as the corresponding transforms along the standard directions explained in Chapter 2. Thus, following the notation introduced in Section 2.4, we denote as $\text{S-AWT}(\mathbf{M}_\Lambda, n_1, n_2)$ the skewed anisotropic transform built on the lattice Λ that has n_1 and n_2 transforms in one iteration step along the transform and alignment directions, respectively. We call the basis functions of the S-AWT *directionlets* since they are anisotropic and have a specific direction. Similarly, we denote the skewed standard WT as S-WT and the skewed FSWT as S-FSWT. The corresponding basis functions are shown in Figure 3.5 for the directions along the vectors $\mathbf{d}_1 = [1, 1]$ and $\mathbf{d}_2 = [-1, 1]$. Notice that the skewed transforms are applied in all cosets of the lattice Λ separately.

The basis functions of the skewed transforms have DVM in *any* two directions with rational slopes. Recall that the N th order DVM along the direction with a rational slope $r_1 = b_1/a_1$ is equivalent to requiring the z -transform of a basis function to have a factor term $(1 - z_1^{-a_1} z_2^{-b_1})^N$ [27; 28; 103]. The following lemma gives the number and directions of the DVM in directionlets.

Lemma 3.4 *Assume that the directionlets of the $\text{S-AWT}(\mathbf{M}_\Lambda, n_1, n_2)$ are obtained using a 1-D wavelet with N vanishing moments. Then, at each scale of the iteration, there are:*

- (a) $2^{n_1} - 1$ directionlets with the N th order DVM along the transform direction of the lattice Λ ,
- (b) $2^{n_2} - 1$ directionlets with the N th order DVM along the alignment direction of the lattice Λ , and
- (c) $(2^{n_1} - 1)(2^{n_2} - 1)$ directionlets with the N th order DVM along both directions.

Proof: Recall first from [103] that 1-D filtering using the filter $H(z)$ along the transform direction of the lattice Λ is equivalent to filtering in the 2-D discrete space using $H(z_1^{a_1} z_2^{b_1})$. Similarly, filtering along the alignment direction of the lattice Λ is equivalent to filtering in

Table 3.1: Orders of approximation by the S-WT, S-FSWT and S-AWT (directionlets) built on the lattice Λ determined by \mathbf{M}_Λ applied on the class S-Mondrian($\mathbf{M}(r_1, r_2), k_1, k_2$).

	$\mathbf{M}_\Lambda = \mathbf{M}(r_1, r_2)$	$\mathbf{M}_\Lambda \neq \mathbf{M}(r_1, r_2)$
S-WT	$(k_1 + k_2)M$	$(k_1 + k_2)M$
S-FSWT	$(k_1 + k_2)(\log_2 M)^2$	$(k_1 + k_2)M$
S-AWT	$(k_1 a + k_2/a)M$	$(k_1 + k_2)M$

the 2-D discrete space using $H(z_1^{a_2} z_2^{b_2})$. Since the 1-D HP filter has N vanishing moments, its z -transform has a factor term $(1 - z^{-1})^N$. Therefore, the HP filtering along the transform and alignment directions uses the equivalent filters with the factor terms $(1 - z_1^{-a_1} z_2^{-b_1})^N$ and $(1 - z_1^{-a_2} z_2^{-b_2})^N$, respectively, in the z -transforms.

Filtering using the 1-D two-channel filter-bank along two directions in the construction of the S-AWT (see Figure 2.5(a)) yields (a) $2^{n_1} - 1$ subbands with HP filtering along only the transform direction, (b) $2^{n_2} - 1$ subbands with HP filtering along only the alignment direction, and (c) $(2^{n_1} - 1)(2^{n_2} - 1)$ subbands with HP filtering along both directions. Thus, the statement of the lemma follows directly. \square

Efficiency of representation of the class S-Mondrian($\mathbf{M}(r_1, r_2), k_1, k_2$) by the three skewed transforms depends on matching between the directions of discontinuities and the directions used in these transforms. If these directions are matched, then the orders of nonzero coefficients in band-pass subbands are equal to the orders calculated in Chapter 2 (see Table 2.1). Otherwise, they are given by the result in Lemma 3.3. The following lemma formalizes this statement. The proof is omitted since it uses the same arguments as in Lemmas 2.1 to 2.3.

Lemma 3.5 *Given is an $M \times M$ pixel image from the class S-Mondrian($\mathbf{M}(r_1, r_2), k_1, k_2$). The S-WT, S-FSWT and S-AWT($\mathbf{M}_\Lambda, n_1, n_2$) with 1-D wavelets having enough vanishing moments built on the lattice Λ determined by the generator matrix $\mathbf{M}_\Lambda = \mathbf{M}(r_1, r_2)$ give $O((k_1 + k_2)M)$, $O((k_1 + k_2)(\log_2 M)^2)$ and $O((k_1 a + k_2/a)M)$ nonzero coefficients in band-pass subbands, respectively. Here, $a = (2^{n_2} - 1)/(2^{n_1} - 1)$.*

The transforms of the image shown in Figure 3.1(a) are given in Figure 3.1(b)-(d). The applied transforms are S-WT, S-FSWT, and S-AWT($\mathbf{M}_\Lambda, 2, 1$), where $\mathbf{M}(r_1, r_2) = \mathbf{M}_\Lambda$. Table 3.1 summarizes the orders of nonzero coefficients in band-pass subbands in the case of both matched and mismatched directions.

Notice that the lattice-based method allows for a more general construction of M-DIR transforms using more than two directions in an arbitrary order. These M-DIR transforms and properties of the corresponding basis functions [97] are outlined in Chapter 4.

3.6 Polyphase Representation

Filtering and subsampling across lattices, as explained in Section 3.4, can be efficiently represented in the polyphase domain. Recall first that a two-channel 1-D filter-bank

$$(H_0(z), H_1(z))$$

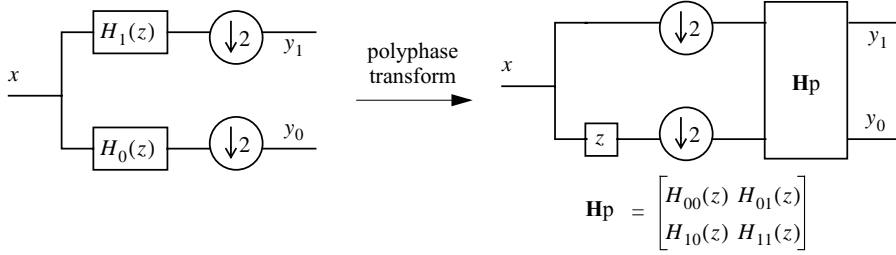


Figure 3.6: A 1-D filter-bank $(H_0(z), H_1(z))$ with the subsampling factor 2 is represented in the polyphase domain with the corresponding polyphase components $H_{00}(z)$, $H_{01}(z)$, $H_{10}(z)$, and $H_{11}(z)$.

followed by a subsampler by the factor 2 can be given in terms of the polyphase components as [101; 104]

$$\begin{aligned} H_0(z) &= H_{00}(z^2) + zH_{01}(z^2) \quad \text{and} \\ H_1(z) &= H_{10}(z^2) + zH_{11}(z^2). \end{aligned}$$

Here, H_{00} , H_{01} , H_{10} , and H_{11} are the polyphase components of the filters $H_0(z)$ and $H_1(z)$ that correspond to even and odd samples of the impulse response, respectively. Such a polyphase representation is shown in Figure 3.6.

Similarly, we can find the equivalent polyphase components of a 2-D filter-bank $(H_0(\mathbf{z}), H_1(\mathbf{z}))$, where $\mathbf{z} = (z_1, z_2)$, applied in the lattice-based method, as explained in Section 3.4. Recall that the filters $H_0(\mathbf{z})$ and $H_1(\mathbf{z})$ used in this method are purely 1-D filters, that is, $H_0(\mathbf{z}) = H_0(z_1)$ and $H_1(\mathbf{z}) = H_1(z_1)$. To illustrate this polyphase decomposition, we consider the particular example with the lattice Λ determined by the generator matrix

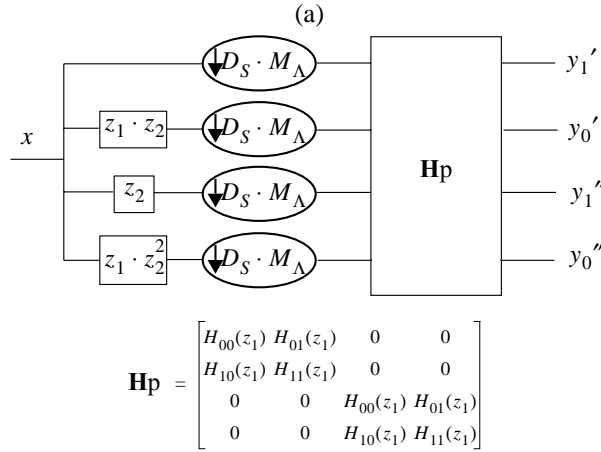
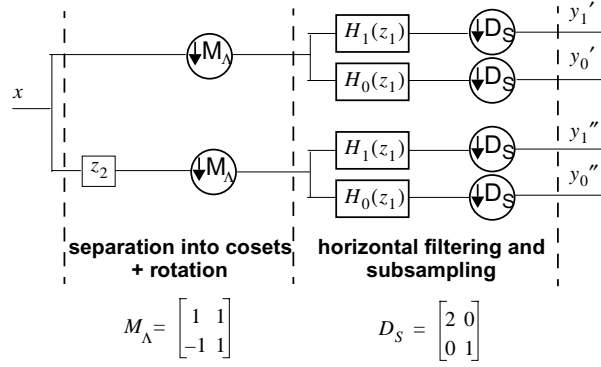
$$\mathbf{M}_\Lambda = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix},$$

as shown in Figure 3.4(a). Recall that the lattice-based filtering and subsampling are applied in each coset of the lattice Λ separately. Thus, the equivalent scheme has two sections, which are (a) separation into two cosets and (b) 1-D filtering and subsampling in the transform direction (Figure 3.7(a)). Notice that filtering in the transform direction is performed as horizontal filtering preceded by rotation by the generator matrix \mathbf{M}_Λ .

Since the total subsampling rate is $|\det(\mathbf{D}_s \cdot \mathbf{M}_\Lambda)| = 4$, the polyphase representation of such a filter-bank consists of 4 polyphase components. The equivalent polyphase representation is shown in Figure 3.7(b), where the polyphase transform \mathbf{H}_p is *block-diagonal*, that is,

$$\mathbf{H}_p = \begin{bmatrix} H_{00}(z_1) & H_{01}(z_1) & 0 & 0 \\ H_{10}(z_1) & H_{11}(z_1) & 0 & 0 \\ 0 & 0 & H_{00}(z_1) & H_{01}(z_1) \\ 0 & 0 & H_{10}(z_1) & H_{11}(z_1) \end{bmatrix}.$$

Notice that the block-diagonal polyphase transform with two identical blocks is a consequence of the separable transforms applied across cosets. This property allows for a simple



(b)

Figure 3.7: The lattice-based filtering and subsampling can be represented in the polyphase domain. (a) The 2-D two-channel filter-bank applied in the example shown in Figure 3.4(a). Filtering and subsampling are applied in 2 cosets separately. (b) Equivalent polyphase representation contains 4 components. The polyphase transform \mathbf{H}_p is block-diagonal.

filter design and computational efficiency in the polyphase domain. Such separability in the polyphase domain has also been used in other 2-D filter-bank designs [4; 67].

3.7 Conclusions

In this chapter, we have proposed new lattice-based anisotropic transforms that have the corresponding oriented basis functions with DVM along different directions, not only horizontal and vertical. The transforms retain the separable filtering and subsampling from the standard 2-D WT, while achieving an efficient description of anisotropic oriented features in images. The corresponding polyphase representation of these filtering and subsampling operations is given by a block-diagonal polyphase transform with separable components.

Chapter 4

General Multi-directional Wavelet Transforms and Regularity

4.1 Introduction

Visual 2-D features in real images contain information in more orientations than only horizontal and vertical. The 2-D separable standard or skewed wavelet transforms (WT) are limited to provide a sparse representation of these features because they use only two transform directions in the construction. This has motivated recently an important research effort towards the design of “true” multi-directional (M-DIR) wavelet bases with 2-D functions that have *directional vanishing moments* (DVM) along more directions.

The lattice-based separable filtering and subsampling operations explained in Chapter 3 allows for the construction of skewed transforms along any two directions with rational slopes. However, a similar method can be used to generalize the construction of the transforms allowing for more than only two transform directions.

An important property of the standard wavelet bases is *regularity*, which consists of a smoothness condition on the corresponding continuous wavelet basis functions. Regular wavelet bases are useful in many image processing applications, in particular, image compression. The main reason why regularity plays an important role in a multiresolution decomposition, such as the wavelet transform, is in a better visual perception of the reconstructed images, that is, perturbation of transform coefficients results in a smooth and perceptually less annoying perturbation in the image domain.

In this chapter, we propose a novel design of general M-DIR wavelet bases, which allows *separable* filtering that can be performed using standard regular 1-D wavelet filter banks. The lattice-based construction allows for a large freedom in the 1-D filters and directions that can be iterated in *any* order. Thus, our approach keeps exactly the same complexity as the traditional separable scheme while globally, it defines a powerful and general truly M-DIR transform with DVM along *any* combination of directions. We also address the issue of regularity of these bases, which is a much more complex problem than regularity of the standard 1-D or 2-D wavelet bases.

In the sequel, we explain the construction of the general lattice-based M-DIR wavelet bases that are built using the separable filtering operations and we give two examples of the

M-DIR transform constructions called *cyclic* and *unbalanced*. Then, we analyze regularity of these basis functions.

4.2 General Lattice-based Multi-directional Construction

Recall that the lattice Λ is determined by an integer generator matrix \mathbf{M}_Λ given by (3.3). In order to simplify the notation, we denote the generator matrices of the lattice Λ_j and its sublattice Λ'_j as \mathbf{M}_j and \mathbf{M}'_j , respectively.

The key idea to obtain multi-directionality is to perform *redirection*, where the transform and alignment directions in the i th subsampled lattice Λ'_j are changed. The change of these directions is made by multiplying the generator matrix \mathbf{M}'_j by a full-rank integer redirection matrix \mathbf{T}_j , that is,

$$\mathbf{M}_{j+1} = \mathbf{T}_j \cdot \mathbf{M}'_j = \mathbf{T}_j \cdot \mathbf{D}_s \cdot \mathbf{M}_j. \quad (4.1)$$

Here, the matrix \mathbf{D}_s performs horizontal subsampling and it is given by

$$\mathbf{D}_s = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}. \quad (4.2)$$

The new generator matrix \mathbf{M}_{j+1} defines a new lattice Λ_{j+1} with the new transform and alignment directions. Notice that, if $|\det(\mathbf{T}_j)| = 1$, then the lattice Λ_{j+1} is the same as the lattice Λ'_j , even though the transform and alignment directions are changed. In this case, the redirection is called *unitary*. Otherwise, if $|\det(\mathbf{T}_j)| > 1$, then $\Lambda_{j+1} \subset \Lambda'_j$, and, thus, the lattice Λ_{j+1} partitions the lattice Λ'_j into more cosets [21]. Such a redirection is called *non-unitary*. Notice also that, in the trivial case, when the transform and alignment directions are not changed, the redirection matrix \mathbf{T}_j is equal to the identity matrix \mathbf{I} .

The M-DIR transforms constructed using non-unitary redirections are intuitively less efficient (in terms of providing a sparse representation) because the pixel magnitudes within a sparser coset are less correlated. However, in spite of this inefficiency, non-unitary redirections are necessary when a series of unitary redirections does not allow for transforming in a chosen direction.

To clarify the construction of the M-DIR transforms, we consider an example of a series of 4 transforms along the directions 0° , 90° , 45° , and -45° as shown in Figure 4.1. The first lattice Λ_1 is cubic and it is determined by the identity generator matrix $\mathbf{M}_1 = \mathbf{I}$. The following lattices Λ_2 , Λ_3 , and Λ_4 are determined by the generator matrices

$$\mathbf{M}_2 = \begin{bmatrix} 0 & 1 \\ 2 & 2 \end{bmatrix}, \mathbf{M}_3 = \begin{bmatrix} 2 & 2 \\ -2 & 2 \end{bmatrix}, \text{ and } \mathbf{M}_4 = \begin{bmatrix} -2 & 2 \\ 8 & 0 \end{bmatrix}. \quad (4.3)$$

Notice that the transform directions (determined by the first row-vectors in the generator matrices) in these lattices correspond to the chosen directions 0° , 90° , 45° , and -45° , respectively. From (4.1) it follows that the three redirections, which yield the lattices Λ_2 , Λ_3 , and Λ_4 are given by the redirection matrices

$$\mathbf{T}_1 = \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix}, \mathbf{T}_2 = \begin{bmatrix} 0 & 1 \\ 2 & -1 \end{bmatrix}, \text{ and } \mathbf{T}_3 = \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix}. \quad (4.4)$$

The second redirection determined by the redirection matrix \mathbf{T}_2 is non-unitary since

$$|\det(\mathbf{T}_2)| = 2.$$

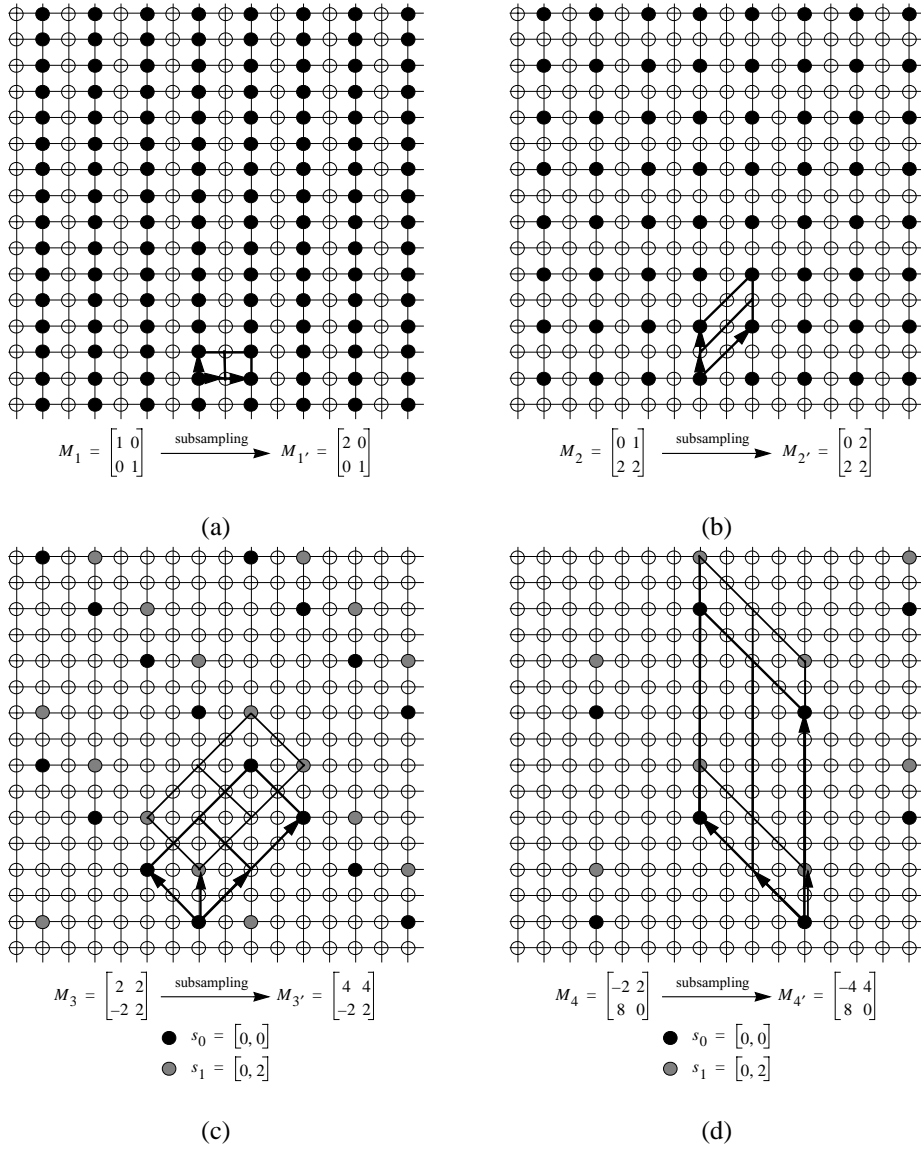


Figure 4.1: Four transform steps are applied along 0° , 90° , 45° , and -45° , respectively. The corresponding lattices (a) Λ_1 , (b) Λ_2 , (c) Λ_3 , and (d) Λ_4 are determined by the generator matrices M_1 , M_2 , M_3 , and M_4 . Filtering and subsampling are applied in the transform direction in each lattice, respectively, determined by the first row-vector in the generator matrices. In order to change the transform and alignment directions, the subsampled lattices determined by the generator matrices $M_{1'}$, $M_{2'}$, $M_{3'}$, and $M_{4'}$ are multiplied by the redirection matrices T_1 , T_2 , and T_3 given by (4.4). The non-unitary redirection shown in (c) partitions the lattice Λ_2' into 2 cosets. The following filtering and subsampling operations are applied in each coset separately.

Thus, the lattice Λ_3 partitions the lattice Λ'_2 into 2 cosets. The following consecutive transforms applied on the lattices Λ_3 and Λ_4 have to be applied in these 2 cosets separately to avoid directional interaction, as described in Chapter 3.

Recall from (4.1) that the redirection matrices have to be full rank and with integer entries resulting in transform and alignment directions with rational slopes. This is the only constraint we have to impose and, thus, a variety of M-DIR transforms is available. Notice that, since the transforms produce 2 subbands (LP and HP) at each scale and the iteration of the transforms is allowed in each subband, the total number of subbands grows exponentially, that is, there are 2^j subbands at the j th scale.

Notice also that the transforms explained in Chapters 2 and 3 are special cases of the M-DIR transforms. Hence, in the standard and skewed WT the only one used redirection matrix is the inverse diagonal matrix given by

$$\mathbf{J} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \quad (4.5)$$

The initial generator matrix \mathbf{M}_1 is the identity matrix \mathbf{I} for the standard WT and given by (3.3) for the skewed WT. Similarly, the fully separable and anisotropic WT (both standard and skewed) can be also represented as the M-DIR transforms with the corresponding redirection matrices equal to either \mathbf{I} or \mathbf{J} . Thus, since these redirections are unitary, the number of cosets in the case of these transforms is retained the same through iterations and that number is equal to the initial number of cosets given by $|\det(\mathbf{M}_1)|$.

Notice that the HP filters in the M-DIR transforms impose vanishing moments in all, not only two, chosen directions producing the corresponding basis functions with combinations of DVM (that is, the functions that have the factor terms $(1 - z_1^{-a_1} z_2^{-b_1})^N$ in the z -transform, for different combinations of a_1 and b_1). Notice also that the transform coefficients are aligned along the alignment directions in the sense given in Lemma 3.2 avoiding directional interactions.

In the continuation we explain two examples of M-DIR constructions, which we call *cyclic* and *unbalanced*.

4.2.1 Cyclic Construction

In order to retain the spatial symmetry of the basis functions and to impose DVM in more directions, we propose the cyclic construction of the M-DIR transforms. In this construction, the transform and alignment directions are iterated in a cyclic manner, that is, for a chosen set of D directions determined by the vectors $\{\mathbf{d}_i\}_{i=1,\dots,D}$, the generator matrices are given by

$$\mathbf{M}_j = \begin{bmatrix} k_1^{(j)} \mathbf{d}_{(j \bmod D)} \\ k_2^{(j)} \mathbf{d}_{((j+1) \bmod D)} \end{bmatrix}, \text{ where } k_1^{(j)}, k_2^{(j)} \in \mathbb{Z}. \quad (4.6)$$

The corresponding redirection matrices satisfy (4.1) and they are given by

$$\mathbf{T}_j = \begin{bmatrix} 0 & 1 \\ m_1^{(j)} & m_2^{(j)} \end{bmatrix}, \text{ where } m_1^{(j)}, m_2^{(j)} \in \mathbb{Z}.$$

The construction is schematically illustrated in Fig. 4.2(a), whereas the basis functions with the Haar and biorthogonal "9-7" 1-D filters are shown in Fig. 4.2(b) and (c).

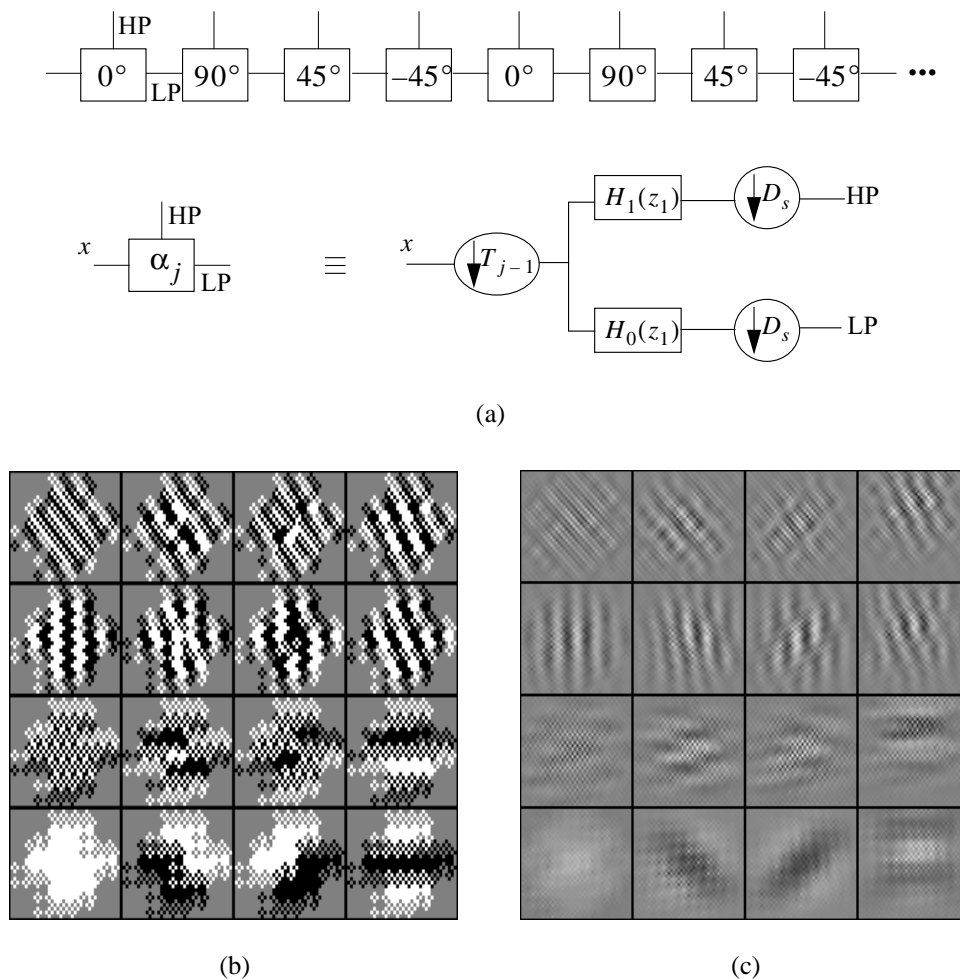


Figure 4.2: The cyclic construction of the M-DIR transforms. (a) Iteration of the transform steps in the cyclic construction is shown only for the LP subbands. Each block implies redirection and transform (filtering and subsampling operations) along the chosen transform directions 0° , 90° , 45° , and -45° . The block-scheme shows the processing in one coset only. The basis functions are obtained in the case of the (b) Haar and (c) biorthogonal "9-7" 1-D filter-banks. DVM are imposed along combinations of the chosen directions.

The cyclic construction is periodic. The following lemma gives the period of the sequence of the redirection matrices and the scaling relation that holds in the sequence of the generator matrices.

Lemma 4.1 *Given a set of D directions determined by the vectors $\{\mathbf{d}_i, i = 1, \dots, D\}$, the sequence of redirection matrices \mathbf{T}_j used in the cyclic transform construction is periodic with the period given by*

$$L = \begin{cases} D, & D \text{ even,} \\ 2D, & D \text{ odd.} \end{cases}$$

Moreover, $\mathbf{M}_j = K\mathbf{M}_{j-L}$, where

$$K = 2^{L/2} \prod_{j=1}^L |\det(\mathbf{T}_j)|^{1/2}. \quad (4.7)$$

Proof: Periodicity of the redirection matrices \mathbf{T}_j follows from the definition of the generator matrices \mathbf{M}_j in the cyclic construction given by (4.6). If D is even, then the generator matrix \mathbf{M}_j is, by definition, the subsampled version of the matrix \mathbf{M}_{j-D} and the redirection matrix \mathbf{T}_j is equivalent to the matrix \mathbf{T}_{j-D} . Otherwise, if D is odd, then the sequence of redirection matrices is repeated after two cycles and, thus, the period is $2D$.

It follows from the recursive relation (4.1) that the factor K is given by the ratio

$$\sqrt{\left| \frac{\det(\mathbf{M}_n)}{\det(\mathbf{M}_{n-L})} \right|}$$

resulting in (4.7). \square

The corresponding cyclic basis functions have DVM along combinations of the D chosen transform directions. The number and distribution of the DVM across subbands is given in the following lemma.

Lemma 4.2 *Given a set of D directions determined by the vectors $\{\mathbf{d}_i, i = 1, \dots, D\}$, there are*

$$\binom{D}{n} = \frac{D!}{n!(D-n)!} \quad (4.8)$$

corresponding cyclic basis functions at one scale that have DVM in n directions, where $0 \leq n \leq D$.

Proof: Recall from the proof of Lemma 3.4 that an N th order DVM imposed in the 2-D filter $H(z_1, z_2)$ along the rational slope direction determined by the vector $\mathbf{d}_i = [a_i, b_i]$ is equivalent to the z -transform of the filter to have the factor term $(1 - z_1^{-a_i} z_2^{-b_i})^N$. This term is imposed by 1-D HP filtering in the direction \mathbf{d}_i .

Since the full decomposition tree is performed in one cycle of the transform, there are 2^D subbands per cycle. The number of subbands that have DVM in n directions is equal to the number of transform sequences in which HP filtering is repeated exactly n times. It follows from combinatorics that this number is given by (4.8). Furthermore, notice that there exists exactly one subband that does not have any DVM (that is, for $n = 0$) and that subband is LP. \square

Notice that the basis functions that correspond to the previous example constructed using 4 transform directions and shown in Figure 4.2 (b) and (c) have DVM along combinations of these 4 directions.

4.2.2 Unbalanced Construction

In the second M-DIR construction, which we call unbalanced, only two transform directions are iterated. In order to get DVM along more than only two directions, the transform steps along two other directions (different than the iterated directions) are performed only once. Notice that, unlike in the case of the cyclic construction, the transform directions are used in an unbalanced manner, that is, the first pair of directions is used more than the second. Notice also that the analysis of regularity of the corresponding basis functions is easier than in the case of the cyclic basis functions because most of the regularity properties is inherited from the two-directional case.

An example of the unbalanced construction with the iterated transform steps along 0° and 90° and a transform step along 45° and -45° is shown in Fig. 4.3(a). The corresponding basis functions with the Haar and biorthogonal "9-7" 1-D filters are shown in Fig. 4.3(b) and (c).

4.3 Regularity of Multi-directional Wavelet Bases

Recall from [101] that a sufficient condition for the convergence in the L_2 sense of an infinite sequence of functions $\{f_n\}_{n=1}^\infty$ to a limit function $f(t)$ is given by

$$\lim_{n \rightarrow \infty} \|f - f_n\|_2 = 0. \quad (4.9)$$

For *regular* iterated 1-D filter-banks, the corresponding iterated continuous basis functions converge through iterations to a piecewise smooth function in the L_2 sense [25; 101]. Furthermore, the Fourier transform of the iterated basis functions also converges to the Fourier transform of the limit function.

Regularity of the iterated filter-banks is equivalent to having an exponential decay of the first-order differences of the filter coefficients through scales (Lipschitz or Hölder exponents) [17; 18; 72; 101], that is

$$\max_n \left| g_0^{(j)}[n+1] - g_0^{(j)}[n] \right| = O(2^{-\alpha j}). \quad (4.10)$$

In the Fourier domain, regularity implies a decay of the Fourier transform of the limit function given by

$$O\left(\frac{1}{(1+|\omega|)^\alpha}\right). \quad (4.11)$$

Recall that the iterated filters and upsamplers can be expressed by one equivalent filter and upsampler using the Noble identities [95]. It has been shown in [25; 101] that the convergence of the Fourier transform of the equivalent scaling and wavelet filters in a wavelet filter-bank is determined by the behavior of the product

$$\prod_{j=1}^n M_0\left(\frac{\omega}{2^j}\right) \quad (4.12)$$

for large n , where $M_0(\omega) = G_0(e^{i\omega})/G_0(1)$ is a normalized version of the LP synthesis filter $G_0(z)$. For a regular filter-bank, the product in (4.12) converges in the L_2 sense as $n \rightarrow \infty$.

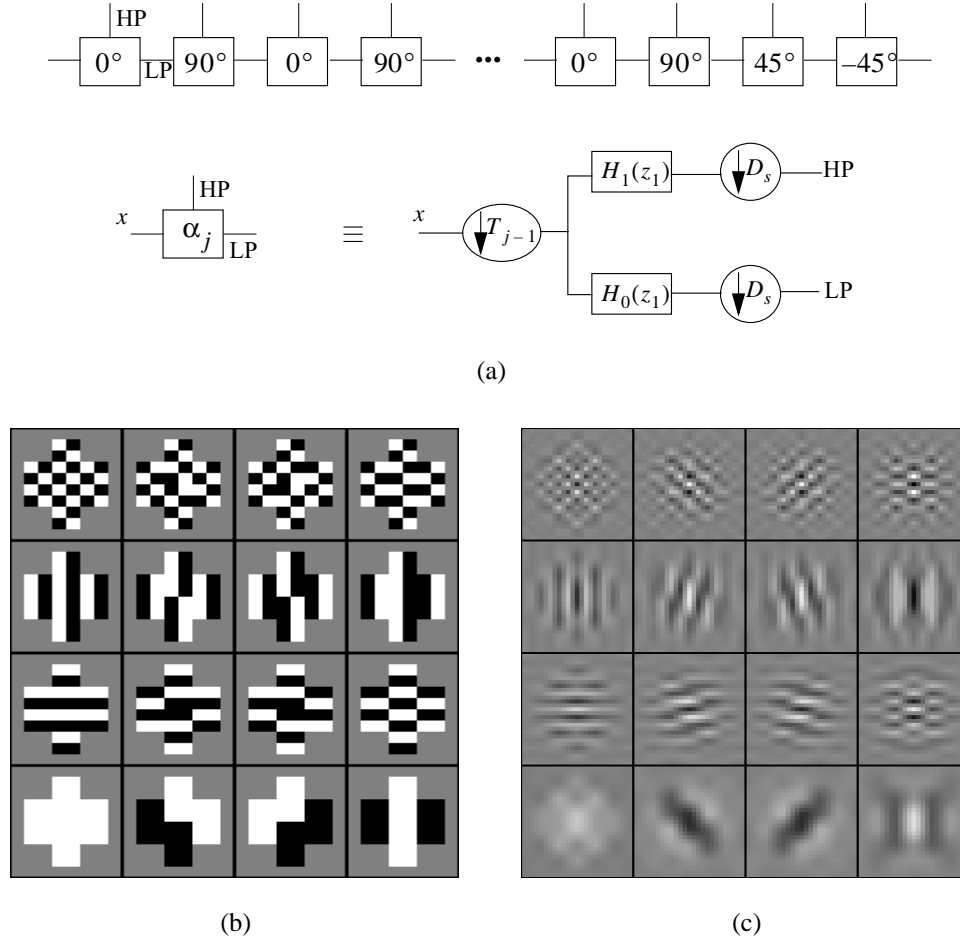


Figure 4.3: The unbalanced construction of the M-DIR transforms. (a) The construction is shown for the LP subbands only. Each block implies redirection and transform (filtering and subsampling operations) in one direction. The transform directions are used in an unbalanced manner, that is, the transform steps along the directions 0° and 90° are iterated and the iteration is followed by one step of the transforms along the directions 45° and -45° . The block-scheme shows the processing in one coset only. The basis functions are obtained using the (b) Haar and (c) biorthogonal "9-7" 1-D filters. DVM are imposed along combinations of the chosen 4 directions.

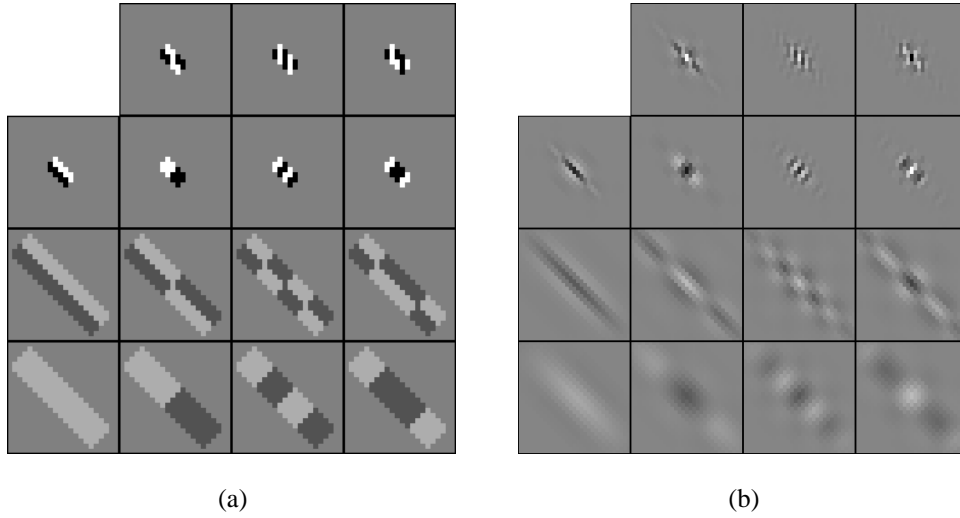


Figure 4.4: The basis functions that correspond to the two-directional transform constructions (directionlets). This transform construction is proposed in Chapters 2 and 3. In this example the directionlets are constructed along the transform directions 45° and -45° for (a) Haar and (b) biorthogonal "9-7" 1-D filter-banks. Both basis functions are regular within cosets.

The analysis of regularity in 2-D is very complex in general. However, if the Fourier transform of the iterated scaling function is separable, then the regularity analysis inherits the properties from the 1-D counterparts, as shown in the continuation.

Notice that the notation of matrix-exponents used in the following analysis is taken from [103] and is also used in [52]. Hence, we define

$$\mathbf{z}^{\mathbf{M}_j^T} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} = \begin{bmatrix} z_1^{a_1} \cdot z_2^{b_1} \\ z_1^{a_2} \cdot z_2^{b_2} \end{bmatrix}.$$

4.3.1 Two-directional Constructions

Recall that the two-directional transform constructions are explained in Chapters 2 and 3. An example of the basis functions is shown in Figure 3.5 for the transform directions 45° and -45° and the two directionlets for the Haar and biorthogonal "9-7" 1-D filter-banks are presented again in Figure 4.4.

Given a 1-D regular wavelet filter-bank, the two-directional transform constructions lead to the basis functions that are regular within cosets. Owing to separability of the construction, the Fourier transforms of the basis functions are also separable and, thus, regularity is inherited from the regularity of the 1-D filter-bank. Furthermore, the limit functions in each coset are the same. This property is formalized in the following theorem.

Theorem 4.1 *Given a regular 1-D wavelet filter-bank, the skewed standard, fully separable, and anisotropic transform constructions result in regular basis functions within cosets, for*

any chosen pair of directions. Moreover, the iterated basis functions in each coset converge to the same limit function.

Proof: Using the Noble identities [95], the equivalent iterated filter is given by

$$G_e(\mathbf{z}) = \prod_{j=1}^n G_0(\mathbf{z}^{\mathbf{M}_j^T}), \quad (4.13)$$

where $\mathbf{z} = [z_1, z_2]^T$. Since the redirection matrices in the case of the standard, fully separable, and anisotropic constructions are separable, the filter $G_e(\mathbf{z})$ can be represented by the product $G_e(\mathbf{z}) = G_{e1}(\mathbf{z}^{\mathbf{d}_1^T}) \cdot G_{e2}(\mathbf{z}^{\mathbf{d}_2^T})$, where the row-vectors $\mathbf{d}_1 = [a_1, b_1]$ and $\mathbf{d}_2 = [a_2, b_2]$ constitute the initial generator matrix \mathbf{M}_1 given by (3.3). Thus, the equivalent product that determines regularity of the basis functions (as given in (4.12)) can be written as follows

$$\prod_{j_1=1}^{n_1} M_0\left(\frac{\mathbf{d}_1 \cdot \mathbf{w}}{2^{j_1}}\right) \cdot \prod_{j_2=1}^{n_2} M_0\left(\frac{\mathbf{d}_2 \cdot \mathbf{w}}{2^{j_2}}\right), \quad (4.14)$$

where the 2-D frequency \mathbf{w} is defined as $\mathbf{w} = [\omega_1, \omega_2]^T$. Recall that both products in (4.14) converge as $n_1, n_2 \rightarrow \infty$ since the corresponding 1-D filters are regular. Therefore, the whole product in (4.14) also converges and a component of the 2-D separable filter in one coset is regular.

Since the construction of the transform is identical in each coset, it directly follows that the corresponding basis functions converge to the same limit function. \square

Notice that if the transform is built using only one coset (that is, if $|\det(\mathbf{M}_1)| = 1$), then the whole filter is regular. Otherwise, the components of the filter in each coset converge separately to the same limit function, but the whole filter is not regular in general since the decay given by (4.10) is not satisfied.

Notice also that the separate construction of directionlets in cosets is similar to the problem of the construction of multiwavelets [80; 88; 102], where a space-variant wavelet filterbank is constructed using a set of different scaling and wavelet functions. In fact, directionlets can be considered as a special case of 2-D multiwavelets, where the scaling and wavelet functions corresponding to each coset are the same.

For the case of the unbalanced M-DIR construction, recall that only two transform directions are iterated. Thus, the analysis of regularity of the basis functions is similar to the two-directional case. The result of Theorem 4.1 still holds since the asymptotic behavior of the equivalent filters, as n grows, also depends on the product (4.13). Hence, the equivalent basis functions are regular within cosets. However, in this case, the limit functions are not necessarily the same in different cosets.

4.3.2 Cyclic Construction

The analysis of regularity in the case of the cyclic construction is much more complex because the 1-D regularity condition given by (4.12) cannot be applied directly as in the case of the two-directional constructions.

In order to clarify this, we consider the equivalent iterated filter $G_e(\mathbf{z})$. It follows from

Lemma 4.1 that the equivalent iterated filter $G_e(\mathbf{z})$ given by (4.13) becomes

$$\begin{aligned} G_e(\mathbf{z}) &= \prod_{j=1}^n G_0(\mathbf{z}^{\mathbf{M}_j^T}) = \prod_{i=1}^L \prod_{j=1}^{\lfloor \frac{n-i}{L} \rfloor} G_0(\mathbf{z}^{\mathbf{M}_{(j-1)L+i}^T}) \\ &= \prod_{i=1}^L \prod_{j=1}^{\lfloor \frac{n-i}{L} \rfloor} G_0(\mathbf{z}^{K^{j-1}\mathbf{M}_i}). \end{aligned} \quad (4.15)$$

However, in this case, the convergence of the Fourier transform of the corresponding iterated scaling function of the filter given in (4.15) does not depend on the behavior of the product (4.12) and, thus, in general, we cannot know if the cyclic basis is regular or not.

Even though we cannot provide a theoretical condition on regularity of the cyclic bases, several experiments give us an intuition that these bases are irregular. Recall from [25] that, for an iterated 1-D LP filter $G_0(e^{i\omega})$ with an N th order zero in $\omega = \pi$, we have

$$M_0(\omega) = \left(\frac{1 + e^{i\omega}}{2} \right)^N R(\omega), \quad N \geq 1.$$

Then, a sufficient condition for regularity of this 1-D wavelet basis is to have [25]

$$\sup_{\omega \in [0, 2\pi]} |R(\omega)| < 2^{N-1}.$$

Recall also that the 1-D filter satisfies the orthogonality condition, that is, it holds

$$|M_0(\omega)|^2 + |M_0(\omega + \pi)|^2 = 1. \quad (4.16)$$

This relation can be rewritten as

$$y^N P(1-y) + (1-y)^N P(y) = 1,$$

where $y = \cos^2(\omega/2) \in [0, 1]$ and $P(1-y) = |R(\omega)|^2$. Daubechies showed [25] that a general solution for the polynomial $P(y)$ to satisfy (4.16) is

$$P(y) = \sum_{j=0}^{N-1} \binom{N-1+j}{j} y^j + y^N Q(y) = P_N(y) + y^N Q(y),$$

where $Q(y)$ is an antisymmetric polynomial with the symmetry point at $y = 1/2$. Notice that for $y = 1/2$ it holds that $Q(1/2) = 0$ and $P_N(1/2) = 2^{N-1}$. Therefore, $\sup_{\omega \in [0, 2\pi]} |R(\omega)| = \sup_{y \in [0, 1]} |P(y)|^{1/2}$ is lower bounded by $|P(y = 1/2)|^{1/2} = 2^{(N-1)/2}$ and, thus, for the orthogonal 1-D filter to be regular, it has to satisfy

$$2^{\frac{N-1}{2}} \leq \sup_{\omega \in [0, 2\pi]} |R(\omega)| < 2^{N-1}. \quad (4.17)$$

However, from the experiments, this condition does not ensure regularity of the corresponding 2-D cyclic bases for the length of the 1-D filters up to 16 using the directions 0° , 90° , 45° , and -45° . An intuitive explanation for the irregularity of these cyclic basis functions

is the fact that the cyclic construction ensures the required decay in the frequency given by (4.11) only in the transform directions, but not in all directions, which is a key condition.

Possible solutions to have both regularity and perfect reconstruction (although not orthogonality) is to relax the condition of orthogonality given by (4.16) and to use biorthogonal 1-D filter-banks. In this case, either the synthesis or the analysis filter-bank can be regular because it is difficult to have regularity in both. However, this topic is left for future work.

4.4 Conclusions

In this chapter, we have presented a construction of the general M-DIR transforms that have DVM along more than only two directions. The transforms retain separability of the filtering and subsampling operations. The analysis of regularity of the corresponding basis functions is very complex and, here, we have shown that regularity within cosets is achievable.

Chapter 5

Non-linear Approximation and Compression of Images using Directionlets

5.1 Introduction

The main task of approximation is to represent a signal by a subset of transform coefficients, while the rest of them is set to zero. The transform can be critically sampled (bases) or oversampled (frames). The approximation with N retained transform coefficients is also called *N-term approximation*. We distinguish between linear approximation (LA) and non-linear approximation (NLA). In the first, the indexes of the retained coefficients are fixed, whereas in the latter, they are adapted to the content of the signal.

Owing to truncation of the coefficients, the approximating signal does not match exactly the original one. The quality of the approximation is commonly measured in terms of *mean-square error* (MSE), that is, for a signal \mathbf{x} and its N -term approximation $\hat{\mathbf{x}}_N$, the MSE is given by $\|\mathbf{x} - \hat{\mathbf{x}}_N\|^2$. Notice that, given a signal \mathbf{x} and its transform $\mathbf{y} = \mathbf{F} \cdot \mathbf{x}$, where \mathbf{F} is a tight frame or an orthogonal basis, we have the following inequality

$$\|\mathbf{x} - \hat{\mathbf{x}}_N\|^2 \leq \frac{1}{A} \|\mathbf{y} - \hat{\mathbf{y}}_N\|^2, \quad (5.1)$$

where $\hat{\mathbf{y}}_N$ corresponds to the truncated version of \mathbf{y} with N retained coefficients, the N -term approximation $\hat{\mathbf{x}}_N$ is given by $\hat{\mathbf{x}}_N = A^{-1} \mathbf{F}^T \cdot \hat{\mathbf{y}}_N$, and A is the frame bound of \mathbf{F} (for more details see Appendix 5.A.1). Equality in (5.1) holds if the transform \mathbf{F} is an orthogonal basis.

In the orthogonal case, the optimal strategy to minimize the MSE is to retain the largest-magnitude transform coefficients [56]. Notice that the MSE decays as the number of retained coefficients (approximants) N grows.

Compression using orthogonal transforms is an extension of NLA that consists of (a) approximation, (b) indexing the retained coefficients, and (c) quantization of those coefficients.¹ Thus, the MSE (in this case also called *distortion*) is affected by the two factors: (a) truncation error due to NLA and (b) quantization error.

¹Some algorithms merge quantization and NLA into a single operation producing an embedded bitstream, like

The asymptotic *rate of decay* of the MSE, as N tends to infinity, is a fundamental approximation property of the transform and this value allows us to compare approximation performance of different transforms. The higher the rate of decay, the more efficient the transform is. Similarly, the rate of decay in compression is defined as the asymptotic behavior of the distortion D , as the bitrate R tends to infinity (this is frequently called *R-D behavior*).

In the continuation we compute the rate of decay that can be achieved by directionlets. Also, we present the optimization tool based on dynamic programming using the Lagrange multipliers [6] applied in the compression algorithm. The algorithm described in the sequel uses the embedded zerotree structures of directionlets and it is a modification of the original zerotree algorithm proposed in [81]. Finally, we show some experiments of compression of synthetic and natural images.

5.2 Achievable Approximation and Compression Rate by Directionlets

Mallat [56] and DeVore [26] showed that, for a 2-D piecewise C^2 smooth signal $f(x_1, x_2)$ with a 1-D C^2 smooth discontinuity curve² (which we call C^2/C^2 signal), the lower bound of the MSE is given by $O(N^{-2})$.

Notice that the standard wavelet transform (WT) is far from optimal since its rate of decay for C^2/C^2 images is $O(N^{-1})$ [35; 56]. Some other adaptive or non-adaptive methods have been shown to improve substantially the approximation power. Curvelets [10; 9; 11] and contourlets [27; 28] can achieve the rate $O(N^{-2}(\log N)^3)$, which is nearly optimal. Furthermore, bandelets [64; 65; 66] and wedgelets [31; 74; 75; 108; 107] have been shown to perform indeed optimally for the C^2/C^2 class of images. However, notice that none of these methods is based on critically sampled filter-banks, which are very convenient for compression. Furthermore, a complex non-separable processing is sometimes required.

As we showed in Section 2 and 3, anisotropy and multi-directionality improve the approximation power of the WT while keeping separability, simplicity, and critical sampling. However, the skewed fully separable WT (S-FSWT) cannot yield a high rate of decay since it fails to provide a sparse representation of C^2/C^2 images. On the other hand, the skewed anisotropic WT (S-AWT) is capable of producing a compact representation, but it is still sensitive to the choice of the transform and alignment directions.

Synthetic (including also C^2/C^2) and natural images have geometrical features that vary over space. Directionality, thus, can be considered as a local characteristic, defined in a small neighborhood. This implies the necessity for *spatial segmentation* as a way of partitioning an image into smaller segments with one or a few dominant directions per segment.

The S-AWT is applied on a segmented image, where the transform and alignment directions are chosen independently in each segment. The transform outperforms the standard WT in both approximation and compression rate of decay of the MSE (i.e. distortion). The following theorem gives the rate of decay for C^2/C^2 images.

Theorem 5.1 *Given a 2-D C^2/C^2 function $f(x_1, x_2)$ and $\alpha = (\sqrt{17} - 1)/2 \approx 1.562$,*

zero-trees [81] or SPIHT [79].

² C^2 smoothness of both 1-D and 2-D functions means that the functions are twice continuously differentiable.

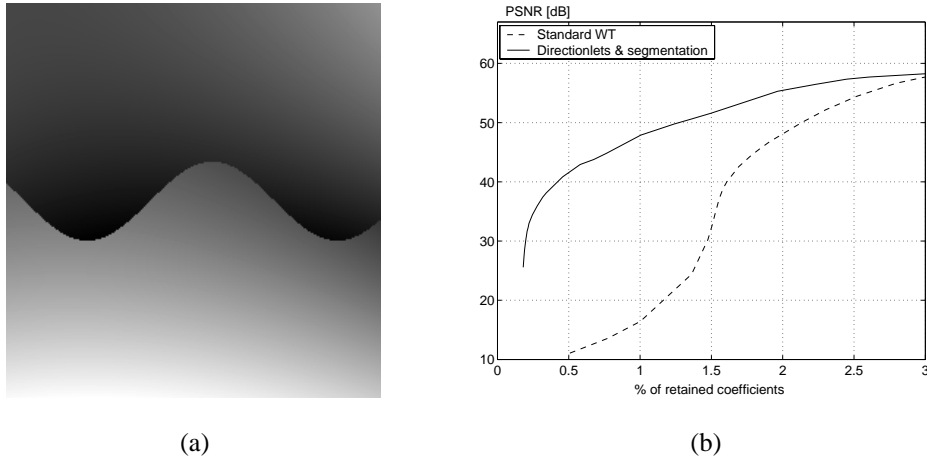


Figure 5.1: An example of NLA of an image from the class C^2/C^2 . (a) An image from the class C^2/C^2 is approximated using the standard WT and the S-AWT($\Lambda,2,1$) with spatial segmentation. (b) The MSE expressed in terms of PSNR is significantly reduced in the case of the S-AWT($\Lambda,2,1$).

(a) The N -term approximation by the S-AWT using spatial segmentation achieves

$$MSE = \|f - \hat{f}_N\|^2 = O(N^{-\alpha}).$$

In that case the optimal anisotropy ratio is $\rho^* = \alpha$.

(b) Compression by the S-AWT, using spatial segmentation and using R bits for encoding, can achieve the distortion D given by

$$D = O(R^{-\alpha}).$$

The proof of the theorem is given in Appendix 5.A.2.

Notice that anisotropic segmentation is used here in the iteration, that is, an image is partitioned into vertical strips of equal widths. The number of segmentation steps depends on the anisotropy ratio, the number of approximants, the number of transform directions, and the first derivative of the C^2 curve (see the proof of Theorem 5.1). In particular, when the optimal anisotropy ratio $\rho^* = \alpha$ is used, the number of segmentation steps does not increase with the number of approximants. However, in reality, because of the discreteness of the transform, this anisotropy ratio cannot be exactly achieved and, in general, the number of segmentation steps has to be increased with the number of approximants. Notice that the S-AWT($\Lambda,3,2$) approximates well the optimal transform³ while retaining iterative segmentation. It follows from the proof of Theorem 5.1 that the number of required transform directions grows with the number of segmentation steps as $O(2^{\beta s})$. Table 5.1 gives the achievable approximation and segmentation rates for the S-AWT($\Lambda,3,2$) and different values of β .

³There are other possible transforms with the anisotropy ratio even closer to optimal but we choose this one for the sake of simplicity.

Table 5.1: Dependence of the approximation rate $\text{MSE} = O(N^{-e_1^*})$ and the number of segmentation levels $s = \eta_1 \log_2(N)$ on the growth rate of the number of transform directions β in the case of the S-AWT($\Lambda, 3, 2$).

β	2	1	0.5	0.25
η_1	1/51	1/26	2/27	4/29
e_1^*	1.55	1.50	1.41	1.24

Although the obtained approximation rate is slower than the ones obtained in [9; 10; 11; 27; 28; 31; 74; 75; 107; 108], we want to emphasize that the S-AWT($\Lambda, 3, 2$) is *critically sampled* and uses only *separable processing*. This is important for compression because, in the case of orthogonal 1-D filter-banks, the Lagrangian optimization-based algorithms still can be applied, making it easier to achieve very good compression.

In order to perform compression, the chosen transform directions in each segment have to be encoded together with the indexes and quantized values of the retained transform coefficients. The bitrate of this overhead information depends on the number of spatial segments and allowed number of transform directions per segment. Recall from Appendix 5.A.2 that the number of spatial segments is equal to 2^s , whereas the number of bits needed to encode the choice of directions in each segment behaves as $O(\log_2(2^{\beta s})) = O(\beta s)$. Thus, the number of overhead bits is given by $R_H = O(\beta s \cdot 2^s)$. However, even though this number grows exponentially with the number of segmentation steps s , the growth rate for the values of β given in Table 5.1 is smaller than the growth rate of the number of indexing and quantization bits and, thus, the dominant asymptotic behavior $D(R)$ remains the same.

Recall also from Section 3.4 that the S-AWT($\Lambda, 3, 2$) is applied in the $|\det(\mathbf{M}_\Lambda)|$ cosets separately. The separate filtering and subsampling in the cosets affect the order of decay of the MSE, but only up to a constant factor and, thus, the rate of decay remains the same.

Figure 5.1 illustrates the gain obtained by NLA using the S-AWT($\Lambda, 2, 1$) with spatial segmentation applied on an image from the class C^2/C^2 when compared to the results of NLA obtained using the standard WT. Furthermore, Figure 5.2 shows an example of the NLA results with a natural image. The image Cameraman shown in Figure 5.2(a) is transformed using the standard 2-D WT without segmentation and the S-AWT($\Lambda, 2, 1$) with segmentation. The MSE obtained by retaining a part of the transform coefficients is presented in Figure 5.2(b). The two reconstructions obtained with 0.98% of retained coefficients for the two methods are shown in Figure 5.2(c) and (d). Finally, the segmentation and adaptation of transform directions for the case in Figure 5.2(d) is illustrated in Figure 5.3.

Notice that a set of S-AWT built using more transform directions and applied on the whole image (without spatial segmentation) yields an overcomplete representation, that is, a tight frame. In this construction, each directionlet is still critically sampled, but overcompleteness is caused by a concatenation of directionlets along more directions (similar to overcompleteness of curvelets). Following the arguments given in [10; 28], we conclude that the best transform in the overcomplete case is S-AWT($\Lambda, 2, 1$) with the vertical alignment direction, since the size of the corresponding directionlets satisfies the parabolic scaling law. However, unlike in [28], where the minimal distance between two transform directions is halved each two scales, here the processing in each direction is made in parallel branches and the number of transform directions must remain the same across scales. Since the redundancy factor is equal to the number of transform directions, the minimal distance between two di-

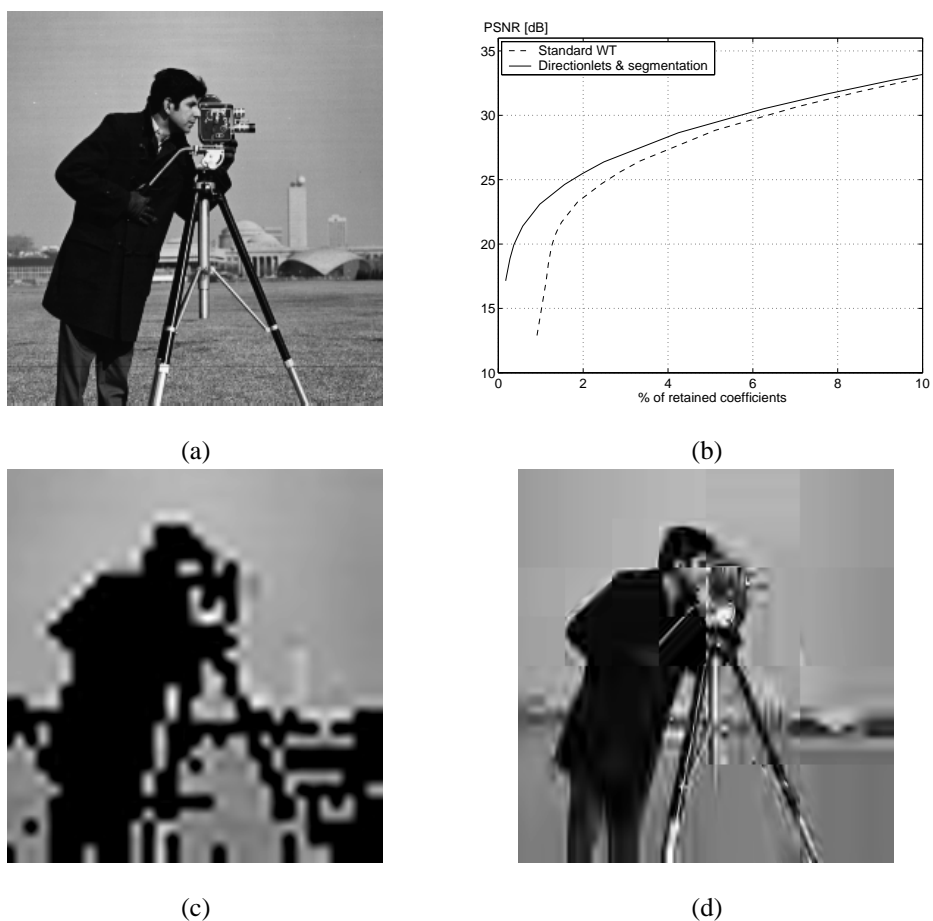


Figure 5.2: An example of NLA of a natural image. (a) The original image Cameraman. The image is approximated using the standard WT and the S-AWT($\Lambda, 2, 1$) with spatial segmentation. (b) The PSNR of the approximated image is significantly improved in the case of the anisotropic transform. (c) The reconstructed image obtained using the standard WT for 0.98% retained coefficients and quality of 13.93dB. (d) The reconstructed image obtained using directionlets with spatial segmentation for the same number of retained coefficients and quality of 23.09dB.

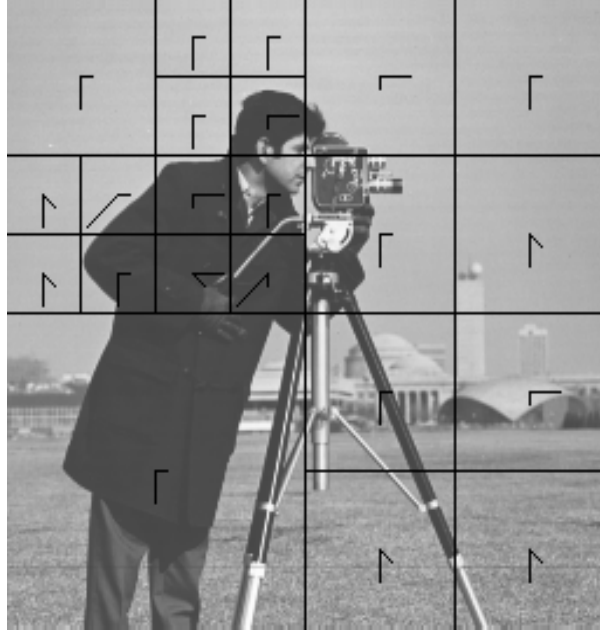


Figure 5.3: The transform directions are adapted to the dominant directions in each segment of the image Cameraman shown in Figure 5.2(a).

rections cannot be arbitrarily small. Thus, the overcomplete directionlets do not provide a good asymptotic order of approximation.

However, this oversampled method is still applicable in practice, when a discrete image is processed. In that case the number of transform scales is finite and upper bounded. The minimal distance between transform directions determines the redundancy factor. Then, the required number of directionlets is used from the beginning and remains the same across scales.

The critically sampled directionlets provide an efficient tool for compression of images. Since the compression algorithm that we propose in this chapter is adaptive to the content of images, it will involve adaptive spatial segmentation and tree pruning. Thus, we review the standard Lagrangian optimization in the next section.

5.3 Lagrange Optimization

The optimization tool based on Lagrange multipliers is explained in details in [70]. Here, we give a brief overview on the use of the tool in the compression algorithm.

The goal of the optimization process is to minimize the average distortion D of a set of signal blocks subject to a total bitrate R_{budget} assuming that the operational R-D curve⁴ is

⁴The R-D curve consists of the operational points in the R-D coordinate system that characterize a compression system.

independent for each signal block and the total bitrate and distortion is obtained as the sum of the bitrates and distortions in each signal block. The second assumption holds only in the case of orthogonal transforms applied in each signal block and, thus, this tool cannot be used in the case of frames and is only approximately true for biorthogonal transforms.

For the i th signal block we denote the chosen quantizer as Q_i and the obtained bitrate and distortion as $R_i(Q_i)$ and $D_i(Q_i)$. This problem is often called resource (bit) allocation, but is more general (e.g. optimal segmentation, choice of different bases, etc.). The problem of independent resource allocation is stated in the form of a constrained minimization, as follows:

$$\min_{Q_i} \sum_{i=1}^N D_i(Q_i), \text{ such that } \sum_{i=1}^N R_i(Q_i) \leq R_{budget}. \quad (5.2)$$

5.3.1 Equivalent Unconstrained Solution

The constrained optimization problem given by (5.2) can be shown to be equivalent to an unconstrained problem given by [70; 82]

$$J_i(\lambda) = \min_{Q_i} \sum_{i=1}^N D_i(Q_i) + \lambda R_i(Q_i). \quad (5.3)$$

Now, the bitrate R_i and the distortion D_i are incorporated into the Lagrangian cost J_i for a given Lagrange multiplier $\lambda \geq 0$. The multiplier trades off distortion for rate. Notice that, since at R-D optimality all signal blocks must operate at the same slope point λ on their R-D curves,⁵ the minimization of the Lagrangian cost J_i is performed *independently* in each signal block. Thus, the optimization problem in (5.3) is unconstrained and substantially simpler than the one in (5.2).

Notice also that different values of the Lagrange multiplier λ correspond to different points on the convex hull (see Figure 5.4). Therefore, λ taking values from 0 to ∞ is equivalent to the operating point sweeping the whole convex hull of the R-D curve. The following theorem formalizes this property. The proof is provided in [82].

Theorem 5.2 *If, given a fixed value λ , the quantizer Q_i^* is the solution to the unconstrained problem of (5.3), then it is also the solution to the constrained problem of (5.2) for the particular case of $R_{budget} = \sum_{i=1}^N R_i(Q_i^*)$.*

Notice that, given a fixed value of λ , the solution to (5.3) is identical to the solution to (5.2) only if they both use the same target R_{budget} . In general, due to the discrete nature of the bitrate, one has to use a convex hull approximation of R_{budget} in the solution to (5.3).

Each value of the approximation of R_{budget} used in (5.3) corresponds to a different value of the multiplier λ . Given a target R_{budget} , the optimal constant slope λ^* is not known *a priori*. However, even though the search for the corresponding λ^* increases the complexity of the optimization algorithm, it can be found at a low computational cost using the bisection algorithm [69; 71; 82], which we present briefly in the sequel.

⁵Otherwise, one can redistribute bits among signal blocks in such a way that the signal blocks with larger slope λ gets more bits and, thus, contribute less to the total distortion. This redistribution can be continued until the optimum is reached, that is, until all signal blocks work at the same slope λ .

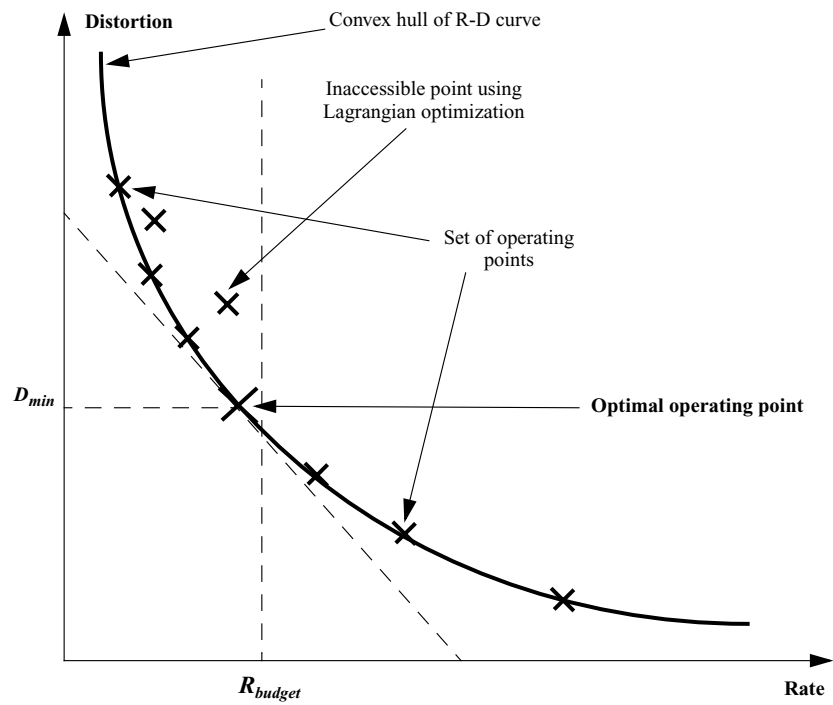


Figure 5.4: Typical R-D characteristic of a compression system. Locations of the operating points depend on various system parameters, like transform choice, quantizer choice, or entropy coder choice. The shown curve is the convex hull of the R-D curve that consists of all optimal (in the R-D sense) operating points. Those points can be reached using Lagrangian optimization techniques. However, some points, which lie above the convex hull, may not be accessible by the fast Lagrangian methods.

5.3.2 Finding the Optimal λ^*

We denote the bitrate and distortion corresponding to the optimal quantizer for the signal block i , which solves the unconstrained optimization problem (5.3), as $R_i^*(\lambda)$ and $D_i^*(\lambda)$, respectively. The algorithm performs a binary search exploiting the monotonicity of $R_i^*(\lambda)$ (recall that the bitrate $R_i^*(\lambda)$ decreases as λ increases).

Assume that we have chosen two initial slopes λ_l and λ_u that satisfy the relation:

$$\sum_{i=1}^N R_i^*(\lambda_u) \leq R_{budget} \leq \sum_{i=1}^N R_i^*(\lambda_l).$$

Notice that if such λ_l and λ_u do not exist, then the given problem is unsolvable, that is, the given R_{budget} is inconsistent for the given quantizers. A trivial choice for λ_l and λ_u would be $\lambda_l = 0$, $\lambda_u = \infty$. We present in Algorithm 5.1 the bisection algorithm in the form as outlined in [70].

Algorithm 5.1 A binary search for the optimal λ^* for a given R_{budget} .

Step 1: Pick $\lambda_l \leq \lambda_u$ such that

$$\sum_{i=1}^N R_i^*(\lambda_u) \leq R_{budget} \leq \sum_{i=1}^N R_i^*(\lambda_l).$$

If the inequality above is an equality for either slope value, stop. We have an exact solution. Otherwise, proceed to **Step 2**.

Step 2: The slope is updated:

$$\lambda_{next} \leftarrow \left| \frac{\sum_{i=1}^N D_i^*(\lambda_l) - D_i^*(\lambda_u)}{\sum_{i=1}^N R_i^*(\lambda_l) - R_i^*(\lambda_u)} \right| + \epsilon,$$

where ϵ is a vanishingly small positive number chosen to ensure that the lower bitrate point is picked if λ_{next} is a singular slope value (i.e. corresponds to a convex hull face slope).

Step 3: Solve (5.3) for λ_{next} .

- If $\sum_{i=1}^N R_i^*(\lambda_{next}) = \sum_{i=1}^N R_i^*(\lambda_u)$, then stop; $\lambda^* = \lambda_u$;
 - else if $\sum_{i=1}^N R_i^*(\lambda_{next}) > R_{budget}$, $\lambda_l \leftarrow \lambda_{next}$; go to **Step 2**;
 - else $\lambda_u \leftarrow \lambda_{next}$; go to **Step 2**.
-

5.4 Anisotropic Zerotree Compression Algorithm

The compression algorithm provided in this section exploits the idea of embedded zerotree structures of coefficients of the standard 2-D WT originally proposed in [81]. However, since the transforms constructed in Chapter 3 and applied in our compression algorithm are

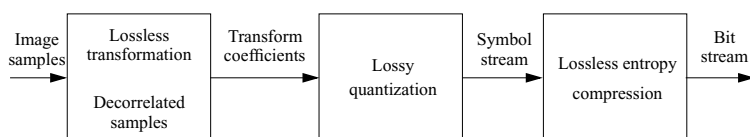


Figure 5.5: A typical transform coder consists of three blocks: transform, quantizer and entropy lossless coder.

anisotropic, the equivalent zerotree structures have to be modified into anisotropic zerotrees. In the sequel, we review the original zerotree-based algorithm and, then, we explain the construction of the anisotropic zerotrees and the optimized compression algorithm that includes spatial segmentation.

Notice that the original Shapiro's zerotree-based compression algorithm is outperformed by several more recent compression algorithms (for instance, SPIHT [79] and SFQ [113; 114] achieve better results than the one presented in [81]). However, these algorithms are based on similar multi-scale-space structures of wavelet coefficients and, therefore, the similar anisotropic grouping can be applied. For that reason, we examine the improvement made by the skewed anisotropic transforms over the standard 2-D WT only in the case of the embedded zerotree algorithm. The work on the corresponding modifications in the SPIHT and SFQ is left for future work.

5.4.1 Standard Embedded Zerotrees

The image coder proposed in [81] consists of three blocks: (1) the standard 2-D WT, (2) quantization, and (3) lossless entropy compression (see Figure 5.5). The goal of the transform implemented in the first block is to produce decorrelated coefficients. Additionally, the mean of the pixels is removed from the original image, so that the transform coefficients can be well modeled as zero-mean, independent.

The goal of quantization is to generate a low entropy stream of symbols so that the symbols can be entropy-coded at some target low bitrate. The quantizers used in the quantization block are symmetric, midtread, and non-uniform.

The total cost of encoding consists of two components: the cost of indexing significant coefficients and the cost of quantization their magnitudes. As the target bitrate decreases, the probability of appearance of insignificant coefficients increases, and the fraction of the encoding cost attributed to the significance map increases.

To improve the efficiency of encoding the significance map of wavelet coefficients, a data structure called *zerotrees* is introduced. A wavelet coefficient x is said to be *insignificant* with respect to a given threshold T if $|x| < T$. The main hypothesis behind the *zerotrees* is that if a wavelet coefficient at a certain scale is insignificant with respect to a given threshold T , then *all* wavelet coefficients of the same orientation in the same spatial location at finer scales are likely to be insignificant with respect to T .

Following this hypothesis, in a hierarchical subband system, with the exception of the finest frequency subbands, every coefficient at a given scale can be related to a set of coefficients at the next finer scale of the same orientation. The coefficient at the coarse scale is called *parent*, and all coefficients corresponding to the same spatial location at the next finer scale of the same orientation are called *children*. Given a parent, the set of all coefficients at

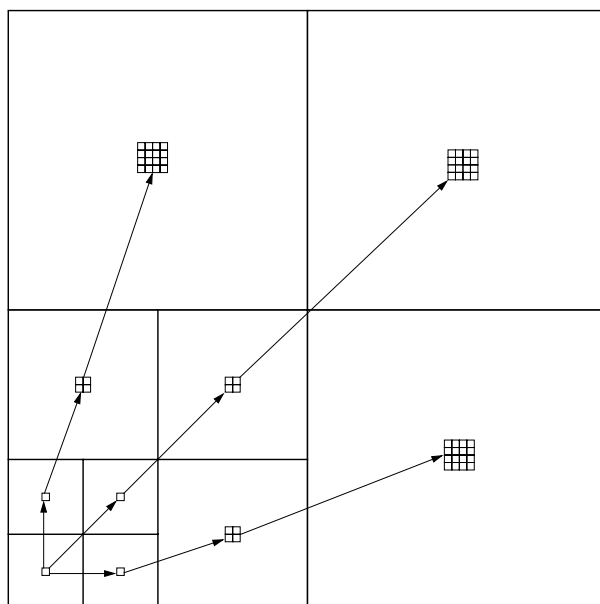


Figure 5.6: Standard zerotrees of wavelet coefficients. Each coefficient, except the ones in the finest subbands, have children - the coefficients at the next finer scale of the same orientation in the same spatial location.

all finer scales of the same orientation corresponding to the same location are called *descendants*. The zerotree structure built of coefficients obtained by the standard 2-D WT is shown in Figure 5.6.

A scanning of the coefficients in the encoding process is performed in such a way that no child is scanned before its parent. The scanning begins at the coarsest scale and progresses to the finest scales (see Figure 5.7).

Given a threshold T a coefficient x is said to be an element of a zerotree if itself and *all* of its descendants are insignificant with respect to T . An element of a zerotree for a threshold T is a *zerotree root* if it is not the descendant of a previously found zerotree root for the same threshold. The encoding of the significance of each coefficient is performed using the three-symbol alphabet: (1) zerotree root, (2) isolated zero (the coefficient is insignificant, but has some significant descendants), and (3) significant with respect to T . Finally, the string of symbols is entropy coded. Notice that the encoding of the finest scale coefficients requires only a 2-symbol alphabet, since the zerotree root symbol is not used.

In practice, to improve embedding, in addition to encoding the significance map, it is useful to encode the sign of significant coefficients along with the significance map. Therefore, four symbols are used: (1) zerotree root, (2) isolated zero, (3) positive significant, and (4) negative significant. The corresponding flow chart is shown in Figure 5.8.

The coefficients that have been found to be significant are further quantized using a successive-approximation quantization (SAQ) approach. The SAQ sequentially applies the thresholds T_0, \dots, T_{N-1} to determine significance, where the thresholds are chosen so that $T_i = T_{i-1}/2$. The initial threshold T_0 is chosen so that the magnitudes of all transform coef-

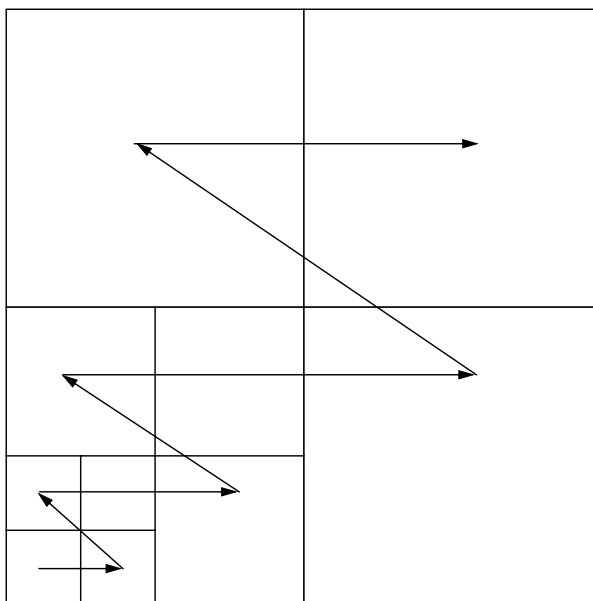


Figure 5.7: The scanning order in the process of encoding the significance of coefficients is made in such a way that no child is scanned before its parent.

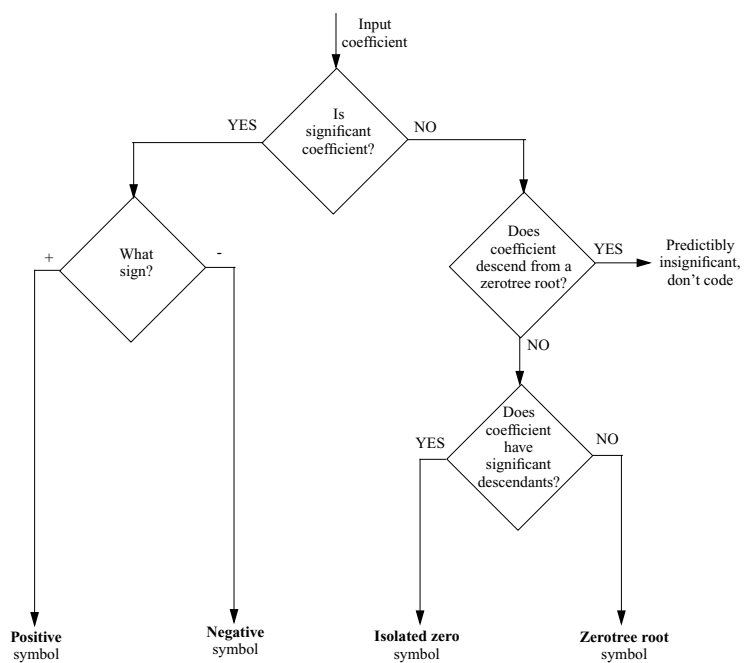


Figure 5.8: The flow chart of encoding a coefficient of the significance map.

ficients are less than $2T_0$. Notice that the SAQ allows for embedded coding, where the quality of image representation grows from coarse to fine when each encoding symbol is sent.

During both the encoding and decoding processes, two separate lists of wavelet coefficients are maintained. The *dominant list* contains the coefficients that have not yet been found to be significant. The coefficients are ordered following the order of scanning shown in Figure 5.7, and within each subband, the set of coefficients is also ordered. Thus, all the coefficients in a given subband appear in the dominant list prior to the coefficients in the next subband. The *subordinate list* contains the magnitudes of those coefficients that have been found to be significant. For each threshold, each list is scanned once.

During a *dominant pass*, coefficients in the dominant list are compared to the threshold T_i to determine their significance. The significant map is then coded using the method shown in Figure 5.8. Each significant coefficient is added to the end of the subordinate list, and the corresponding magnitude in the zerotree of wavelet coefficients is reset to zero so that the future occurrences of a zerotree in the next dominant passes are not prevented.

A dominant pass is followed by a *subordinate pass* in which all coefficients in the subordinate list are scanned and the quantized values of their magnitudes are refined to an additional bit of precision. In this way, the quantization step size, which defines the uncertainty interval, is halved. The encoding is performed using 2 symbols, specifying if the quantized magnitudes fall in the upper or lower half of the previous uncertainty interval. The string of the binary symbols is then entropy coded.

After scanning all the coefficients in the subordinate list, the entries are sorted in descending order of the reconstruction values and the threshold is updated (halved) for the next dominant pass. The process continues to alternate between dominant and subordinate passes until some target criterion is met (the bitrate budget is exhausted or the maximal allowed distortion is achieved).

Algorithm 5.2 summarizes the described encoding method presenting both dominant and subordinate passes.

In the decoding process, each decoded symbol reduces the width of the uncertainty interval. The reconstruction value is chosen to be in the center of the uncertainty interval minimizing the MSE. Notice that the corresponding decoding algorithm is similar to Algorithm 5.2 with two exceptions: the input and output streams of symbols are exchanged and the reconstruction values have to take into account the sign of the decoded coefficients.

The encoding symbols are entropy coded, as depicted in Figure 5.5. The entropy arithmetic coder is used because it is more efficient in the case of dictionary with small number of symbols than the Huffman coder. The arithmetic coder and decoder are implemented as in [111].

Notice also that the stream of encoding symbols can be stopped at any moment. All the lower-rate encoding streams are embedded in the produced stream. Furthermore, the R-D curve can be computed on-line at a high resolution. This property is very convenient for the application in the Lagrangian optimization-based compression algorithm, as we explain in Section 5.4.3.

5.4.2 Skewed Anisotropic Zerotrees

Zerotree structures proposed in [81] hierarchically order wavelet coefficients produced by the standard 2-D WT. The main principle of grouping coefficients in a zerotree is that they have to be located in the same spatial location and belong to subbands of the same orientation.

Algorithm 5.2 Standard zerotree-based encoding algorithm proposed in [81].

Step 1: Apply the 2-D WT on the input image.

Step 2: Initialization

- $T \leftarrow \max_i |x_i|/2 + \epsilon$;
- Add all the coefficients in the LP subband and the coefficients in the subbands at scales $j = 1, \dots, J$ and orientations $d = 1, 2, 3$ to the dominant list in the defined order.

Step 3: Dominant pass

For each coefficient x_i in the dominant list:

- If $|x_i| \geq T$, then:
 - If $|x_i| \geq 0$, then send a **positive** symbol, otherwise send a **negative** symbol; stop if the target condition is met;
 - Append $|x_i|$ to the end of the subordinate list with the reconstruction value equal to $1.5 \cdot T$;
 - Reset the magnitude of the corresponding wavelet coefficient to zero.
- Else, if the coefficient already belongs to a zerotree, then do not code it;
- Else, if the coefficient has significant descendants, then send an **isolated zero** symbol, otherwise send a **zerotree root** symbol; stop if the target condition is met.

Step 4: Subordinate pass

- For each entry in the subordinate list:
 - If the magnitude of the coefficient $|x_i|$ is larger than its reconstruction value, then send **1** and add $T/4$ to the reconstruction value;
 - Else send **0** and subtract $T/4$ from the reconstruction value;
 - Stop if the target condition is met.
- Sort the subordinate list in descending order of the reconstruction values.

Step 5: $T \leftarrow T/2$; go to **Step 3**.

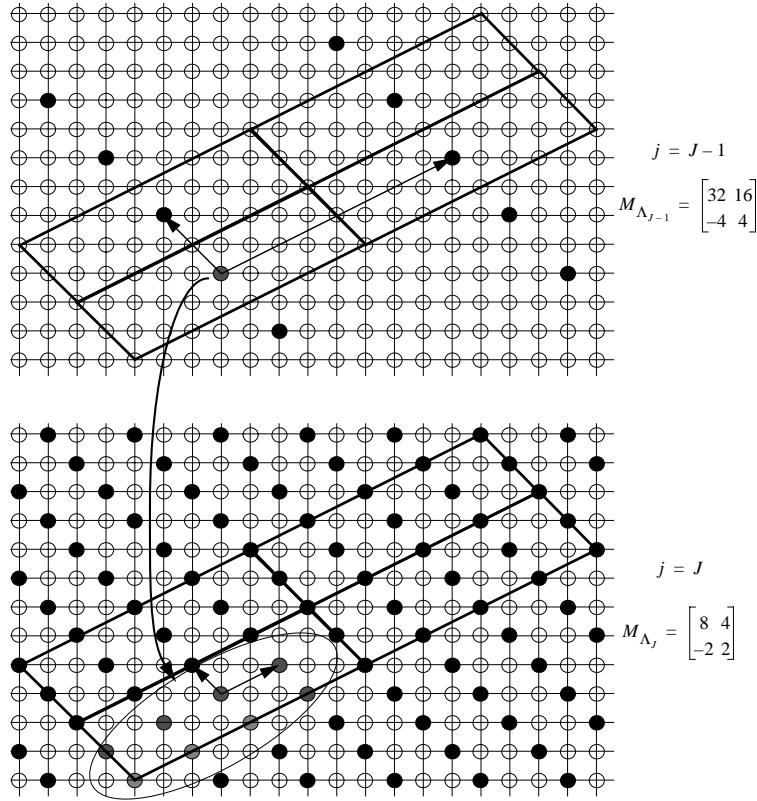


Figure 5.9: The subbands at the coarser scale $j = J-1$ are defined across the lattice Λ_{J-1} with the generator matrix $M_{\Lambda_{J-1}}$. Each coefficient has a set of children at the next finer scale $j = J$. The children belong to the same orientation and coset as their parent and are located in the parallelepiped around the spatial location of their parent.

The S-AWT proposed in Chapter 3 allows for similar zerotree structures. The main principle of grouping directionlets is retained, but, now, the number of children that each coefficient can have may vary depending on the anisotropy ratio of the transform.

Since the S-AWT is defined using lattices, this concept is also convenient to define the corresponding anisotropic zerotrees. Assume that the J -scale S-AWT(M_{Λ_J}, n_1, n_2) is applied on an image, where $M_{\Lambda_J} = [\mathbf{d}_1, \mathbf{d}_2]^T$ is given by (3.3). The transform is applied in each $|\det(M_{\Lambda_J})|$ cosets independently. From the construction of the AWT, as explained in Section 2.4, it follows that the number of HP subbands produced at each scale is $2^{n_1+n_2} - 1$. Similarly to the case of the standard zerotrees, these subbands are said to have different orientations. Therefore, given a S-AWT, there are $2^{n_1+n_2} - 1$ orientations.

Consider two subbands of the same orientation at two consequent scales. We denote the index of the coarser scale as j and that of the finer scale as $j+1$, where $j = 1, \dots, J-1$. The

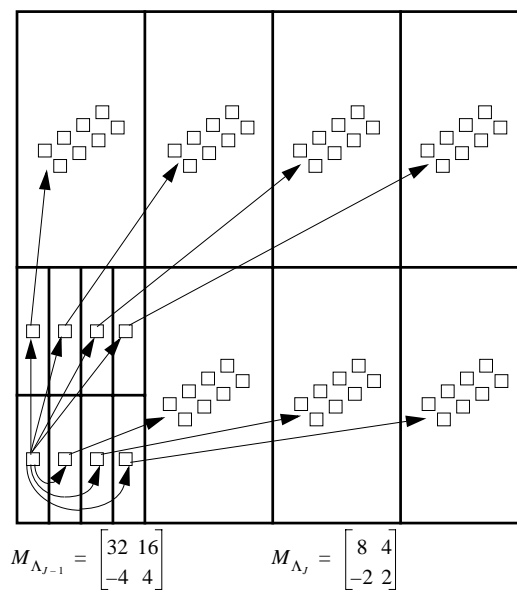


Figure 5.10: The hierarchical structure of the skewed anisotropic zerotrees is similar to the standard structure shown in Figure 5.6. However, now, the number of children that each parent can have depends on the anisotropy ratio of the applied transform.

two subbands are defined across the corresponding lattices given by the generator matrices

$$\mathbf{M}_{\Lambda_j} = \begin{bmatrix} 2^{(J-j+1)n_1} \cdot \mathbf{d}_1 \\ 2^{(J-j+1)n_2} \cdot \mathbf{d}_2 \end{bmatrix}, \text{ and } \mathbf{M}_{\Lambda_{j+1}} = \begin{bmatrix} 2^{(J-j)n_1} \cdot \mathbf{d}_1 \\ 2^{(J-j)n_2} \cdot \mathbf{d}_2 \end{bmatrix}.$$

Notice that the lattice Λ_j is a sublattice of the lattice Λ_{j+1} and recall that the coefficients in the two subbands are separated into $|\det(\mathbf{M}_{\Lambda_j})|$ cosets. A coefficient in the c th coset of the subband at the coarser scale j is parent of a set of children that belong to the same coset c in the subband at the next finer scale $j + 1$. The children are located within the parallelepiped around the parent defined by

$$\alpha_1 \cdot 2^{(J-j+1)n_1} \cdot \mathbf{d}_1 + \alpha_2 \cdot 2^{(J-j+1)n_2} \cdot \mathbf{d}_2, \text{ for } -\frac{1}{2} \leq \alpha_1, \alpha_2 < \frac{1}{2}.$$

The corresponding parent-children relation is illustrated in Figure 5.9 for the case of S-AWT($\mathbf{M}_\Lambda, 2, 1$), where the generator matrix \mathbf{M}_Λ is given by (3.3) for $\mathbf{d}_1 = [2, 1]$ and $\mathbf{d}_2 = [-1, 1]$. The corresponding hierarchical structure of subbands across scales is shown in Figure 5.10 for the same transform.

Notice that the standard zerotree structure is a special case of the skewed anisotropic zerotree structure for $\mathbf{M}_{\Lambda_j} = \mathbf{I}$, where \mathbf{I} is the identity matrix, and $n_1 = n_2 = 1$.

5.4.3 Optimized Compression Algorithm with Spatial Segmentation

In this section, we propose a compression algorithm based on the skewed anisotropic zerotrees of directionlets and quad-tree spatial segmentation. The algorithm performs optimization in the R-D sense using dynamic programming based on the Lagrangian multiplier tool, as explained in Section 5.3.

The motivation for spatial segmentation, as mentioned in Section 5.2, is in the locality of directionality in images. Transform and alignment directions are chosen adaptively to the content of each spatial segment, resulting in a sparser representation of the whole image. In this algorithm, the quad-tree segmentation is implemented, as explained in the sequel.

First, the input image is divided into four equally sized segments in a step of the quad-tree spatial segmentation. The segmentation steps are iterated on each segment until the predetermined maximal segmentation depth (or the minimal size of segments) is reached. The iteration results in a full spatial quad-tree, as shown in Figure 5.11, where each node represents a spatial segment. Notice that each node in the quad-tree have four *children* except the nodes in the last level. The nodes without children are called *leaves*, whereas the ones that have children are called *parents*. Notice also that other spatial segmentation methods are possible, but we use the quad-tree segmentation as the simplest in terms of coding efficiency, since 1 bit per node suffices to encode a path through the tree. The quad-tree segmentation has already been used in other image compression techniques, like in [83; 113; 114]. Also, a similar analysis of dynamic programming in the case of trees is given in [6]. The examination of the impact of more complex segmentation methods on our compression algorithm is left for future work. Some analysis of the multi-tree spatial segmentation can be found in [45; 46; 47].

The optimization process applied on the full spatial quad-tree is divided into 3 embedded phases:

- I Choosing the best length of the encoding bit stream for each node,

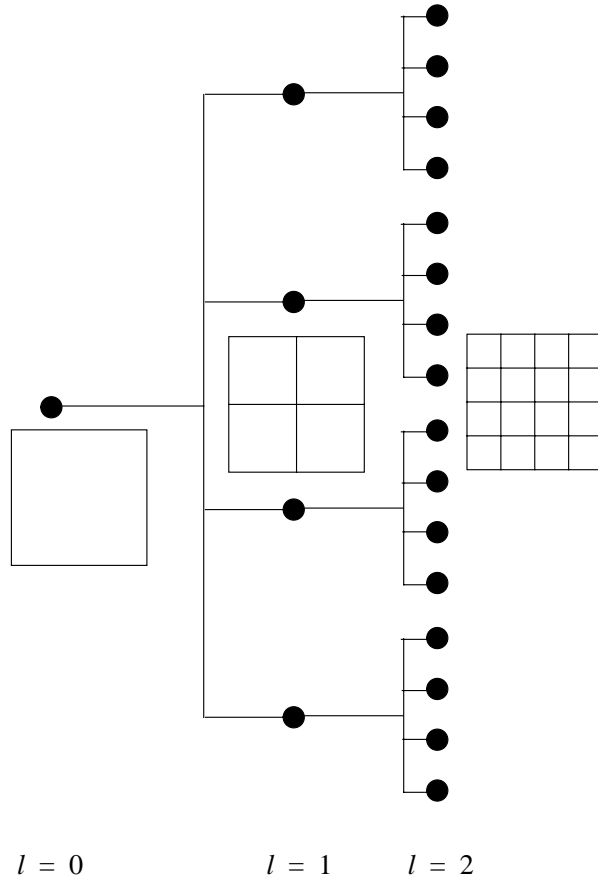


Figure 5.11: The nodes in the full quad-tree represent spatial segments. Maximal depth of the shown spatial quad-tree is 2.

II Exhaustive search for the best pair of transform and alignment directions for each node,

III Pruning the full spatial quad-tree.

In each phase, the Lagrangian cost is minimized for a given constant multiplier λ .

The S-AWT is applied on the spatial segments associated to each node in the full quad-tree shown in Figure 5.11. The transform and alignment directions are combined from a predetermined set of directions in such a way that the corresponding lattice used in the transform is of full rank (i.e. the vectors that approximate the chosen directions are linearly independent; revisit Section 3.4 for more details on the construction of the S-AWT). Notice that a set of D directions generates, in general, $D(D - 1)$ different combinations for the S-AWT. An exception happens in the case when an isotropic transform is applied (for the anisotropy ratio $\rho = 1$) generating $D(D - 1)/2$ different combinations. Then, for each combination of directions, the corresponding transform coefficients are encoded using the zerotree-based encoding algorithm outlined in Algorithm 5.2. The encoding process terminates when the

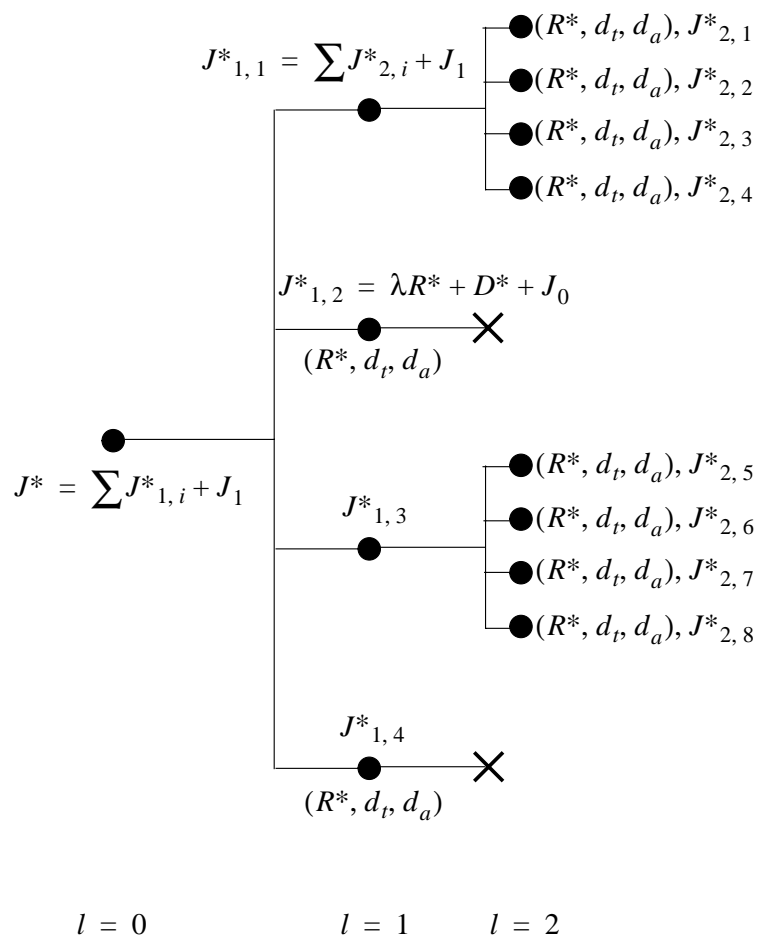


Figure 5.12: An example of a pruned quad-tree is shown. Children are pruned if the sum of their Lagrangian costs is larger than the cost of their parent.

length of the bit stream reaches the predetermined maximally allowed bit budget. For all spatial segments and combinations of directions the maximal bit budget is the same in terms of *bits-per-pixel* (bpp).

For each spatial segment and combination of directions, during the encoding process, the distortion is simultaneously measured and the corresponding R-D curve is recorded. The resolution of the measured R-D curve can be very high (in the extreme case, the distortion can be measured for each bit added to the bit stream). However, in practice, the R-D curve is not perfectly convex at this highest resolution. Since convexity of the curve is a condition for a successful Lagrangian optimization, we measure the distortion at a lower resolution, that is, after each 16 encoding bits.

In the optimization phase I, given a Lagrange multiplier λ , the operating point along the R-D curve that minimizes the Lagrangian cost $J = \lambda R + D$ is chosen. The corresponding optimal bitrate is the allowed target bitrate associated to the particular spatial segment and combination of directions. Notice that a set of minimized Lagrangian costs corresponding to each combination of directions used in the transform is associated to each spatial segment.

In the following phase II, for each spatial segment, the minimal cost in the set is chosen and the winning transform and alignment directions and the corresponding optimal bitrate are associated to the spatial segment. The minimal cost is said to be the cost of the segment (i.e. of the node in the quad-tree).

Finally, in the optimization phase III, the full quad-tree is pruned according to the costs of each node. Children of a node are pruned if [83]

$$J_p + J_0 \leq \sum_{i=1}^4 J_{c_i} + J_1, \quad (5.4)$$

where J_p is the cost of the parent, J_{c_i} are the costs of the children, and J_0 and J_1 represent the costs of encoding the pruning decision (the messages: "prune" and "retain" the children, respectively). In the particular case of the quad-tree, 1 bit per node is used to encode this information and, thus, $J_0 = J_1 = \lambda$. If the children are pruned, then the cost of the parent is updated by adding J_0 . Otherwise, if the children are retained, then the cost of the parent is equal to the sum of the costs of the children and J_1 . The updated cost of the parent is used further in pruning the nodes at upper levels. An example of a pruned quad-tree is shown in Figure 5.12.

The whole encoding algorithm is summarized and outlined in Algorithm 5.3. Notice that, given a Lagrange multiplier λ , the total bit budget is the sum of two components: (1) the sum of the bitrates spent for zerotree encoding of the transform coefficients in leaves of the quad-tree and (2) the bitrate allocated to encoding the quad-tree structure (i.e. the pruning decisions). Since these bitrates depend on the R-D curves in nodes of the quad-tree, the total bit budget is not known a priori. In order to reach the target total bit budget, the bisection algorithm is applied, as explained in Section 5.3.2 and Algorithm 5.1.

5.5 Simulation Results

In this section, we explain the details of the implementation of the compression algorithm outlined in Section 5.4.3. Then, we present some results of the compression of synthetic and natural images.

Algorithm 5.3 Optimized anisotropic zerotree-based encoding algorithm with spatial segmentation.

Step 1: Full quad-tree

If the current spatial segmentation level is less than maximally allowed, then:

- Divide into 4 equally sized segments;
- Repeat **Step 1** for each segment.

Step 2: S-AWT

For all $d_1, d_2 = 1, \dots, D$, such that $d_1 \neq d_2$, apply the S-AWT using d_1 th and d_2 th direction from the predetermined set as the transform and alignment directions.

Step 3: Zerotree encoding - Phase I

For each node in the quad-tree generated in **Step 1** and each transform from **Step 2**:

- Run the zerotree encoding algorithm (outlined in Algorithm 5.2) with anisotropic zerotrees and the maximal bit budget specified as input;
- Record the operating points at the R-D curve after each 16 bits added to the output bit stream;
- Compute the Lagrangian costs for each measured point at the R-D curve and find the minimum;
- Associate the optimal bitrate and the minimal Lagrangian cost to the particular node and transform.

Step 4: Best pair of directions - Phase II

For each node in the quad-tree:

- Choose the minimal Lagrangian cost among the costs associated to that node and all transforms;
- Associate the winning pair of directions, the minimal cost, and the optimal bitrate to the particular node.

Step 5: Pruning the quad-tree - Phase III

- If the current spatial segmentation level is less than maximally allowed, then repeat **Step 5** for each segment (child);
- Else, if (5.4), then prune the children and $J_p \leftarrow J_p + J_0$;
- Else $J_p \leftarrow \sum_{i=1}^4 J_{c_i} + J_1$.

Step 6: Final encoding

- Encode the pruning decisions in the pruned quad-tree;
 - For each leave encode the optimal transform and alignment directions and generate the bit stream using the optimal bitrate as the target bit budget.
-

5.5.1 Implementation Details

The source code that implements the skewed anisotropic zerotree-based compression of images is fully written in C. The code consists of 4 main parts: (1) transform block, (2) zerotree coder, (3) optimization block, and (4) entropy coder.

- (1) The S-AWT is performed for any rational slope direction. In the simulations, the horizontal, vertical, and the two diagonal (along 45° and -45° degrees) directions are used because the number of cosets in the corresponding lattices in these cases is small (2 cosets for the combination of the two diagonal directions, and 1 coset for the other combinations). Filtering is performed using the linear phase biorthogonal "9-7" wavelet filter-bank [2] and symmetric extension at the borders of the image to reduce the impact of border effects [8]. The number of decomposition levels depends on the size of the image and it is hand-optimized. The most frequently value used in the simulations is 4.
- (2) First, the wavelet coefficients produced by the transform are structured in the skewed anisotropic zerotrees. Owing to the lattice-based subsampling, some coefficients at certain scales "hang" without a parent in the corresponding spatial location. These coefficients are joined as children to the spatially closest coefficients at the next coarser scale. Then, the zerotree coder performs the encoding process, as depicted in Algorithm 5.2. The initial threshold T is chosen to be 1% larger than a half of the maximal magnitude of all transform coefficients and that value is encoded at the beginning of the bit stream assigned to the segment.
- (3) The optimization process is embedded, as explained in Section 5.4.3. The corresponding procedures are recursive and follow the three optimization phases in Algorithm 5.3. The mean value is computed and subtracted from pixels once for the whole image and coded at the beginning of the total bit stream. Even though some spatial segments might have a non-zero mean value, this approach is shown to be more efficient at low bitrates for most images because it is too costly to encode all mean values of each spatial segment separately.
- (4) The implemented entropy coder is taken from [111]. The symbols produced during dominant passes are entropy coded using 4 different histograms depending on the significance of the previously scanned coefficient and the parent (see also [81]). The binary symbols generated in subordinated passes are coded using 1 histogram. For all histograms, the maximal frequency is set to 256, since, for that value, the entropy coder shows the best performance in experiments.

5.5.2 Experiments

First, we show the results of compressing a toy example - a bilevel image that contains a circle (see Figure 5.13(a)). The image is compressed using the isotropic transforms built along directions chosen from the set $\{0^\circ, 90^\circ, 45^\circ, -45^\circ\}$ and the results are compared to the results of the standard zerotree-based compression algorithm at different bitrates. The comparison is shown in Figure 5.13(b). The optimal spatial segmentation and choice of directions for each segment are illustrated in Figure 5.14. Notice that, for each segment, the chosen transform and alignment directions are well adapted to the dominant directions of

arcs of the circle. The "empty" segments (that is, the segments with a constant magnitude) are not further segmented as it would involve an additional cost for encoding the segmentation decisions and chosen directions. The corresponding reconstructions of the circle for the two transforms at the same bitrate $R = 0.0378\text{bpp}$ are shown in Figure 5.15(a) and (b). The resulting distortions are expressed in terms of *peak-signal-to-noise-ratio* (PSNR) factor equal to $20 \log_{10}(255/\sigma_e)$, where σ_e is the standard deviation of the error and they are equal to 32.79dB for the standard WT and 34.35dB for the optimized M-DIR transforms. Not only the distortion of the reconstruction using the M-DIR transforms is smaller, but also the visual quality of the image is improved, since the arc of the circle is sharper. A part of the arc in both reconstructed images is magnified and shown in Figure 5.16(a) and (b).

Then, the anisotropic transforms $\text{S-AWT}(\mathbf{M}_\Lambda, 2, 1)$ are applied along the directions chosen from the same set. In Figure 5.13(c), the results of the compression using the anisotropic transform with the optimal segmentation and the choice of directions, the standard WT, and the anisotropic transforms along the standard directions are shown. The anisotropic M-DIR transform outperforms the others. Moreover, a comparison between the optimal isotropic and anisotropic transforms is illustrated in Figure 5.13(d). The anisotropic transform achieves better compression at very low bitrates, but it is overtaken by the isotropic transform at higher bitrates. The reason for a worse performance of the anisotropic transform at higher bitrates lies in a stronger border effect obtained using the elongated basis functions in small sized segments. Furthermore, isotropy of the compressed object also suggests that the isotropic transform provides a better compression results.

Recall that the $\text{S-AWT}(\mathbf{M}_\Lambda, 3, 2)$ is shown in Section 5.2 to have better approximation performance than the $\text{S-AWT}(\mathbf{M}_\Lambda, 2, 1)$. However, even though it provides a faster asymptotic approximation rate, the support of the corresponding basis functions of the $\text{S-AWT}(\mathbf{M}_\Lambda, 3, 2)$ is too large for an efficient implementation in the zerotree-based compression method. The depth of the zerotree structures is limited to a small number of scales and, therefore, the multi-scale dependence of the coefficients is not properly captured. The smaller size of the support of the basis functions of the $\text{S-AWT}(\mathbf{M}_\Lambda, 2, 1)$ allows for a larger number of iteration steps of the transform and, thus, for a "deeper" and more efficient zerotree structures. This is why we rather use the $\text{S-AWT}(\mathbf{M}_\Lambda, 2, 1)$ in the experiments than the $\text{S-AWT}(\mathbf{M}_\Lambda, 3, 2)$.

Next, we present the compression results of another synthetic, but more complex image. An image from the class C^2/C^2 shown in Figure 5.17(a) is compressed using the isotropic and anisotropic transforms with spatial segmentation and directions chosen from the set $\{0^\circ, 90^\circ, 45^\circ, -45^\circ\}$. A comparison of the results obtained by the optimized method using the isotropic transforms and the standard WT is shown in Figure 5.17(b). The isotropic transform is constantly better than the standard WT for a wide range of bitrates. However, for this image that contains an anisotropic object, the optimized method using the anisotropic transform $\text{S-AWT}(\mathbf{M}_\Lambda, 2, 1)$ is substantially better than both methods that use the standard WT and the isotropic transforms, as can be seen in Figure 5.17(c). A comparison between the optimal results using the isotropic and anisotropic transforms is illustrated in Figure 5.17(d). The optimal segmentation and the choice of transform and alignment directions for this image is shown in Figure 5.18. Notice that the transform directions follow the dominant directions of the edge in each segment that contains a sharp transition. A significant difference in the quality of the reconstructed images obtained using the standard WT and the $\text{S-AWT}(\mathbf{M}_\Lambda, 2, 1)$ is noticeable in Figure 5.19. These reconstructions are made at the same bitrate $R = 0.0263\text{bpp}$ and the resulting distortions are $D = 26.33\text{ dB}$ in the case of the standard WT and $D = 38.62\text{dB}$ in the case of the anisotropic transform.

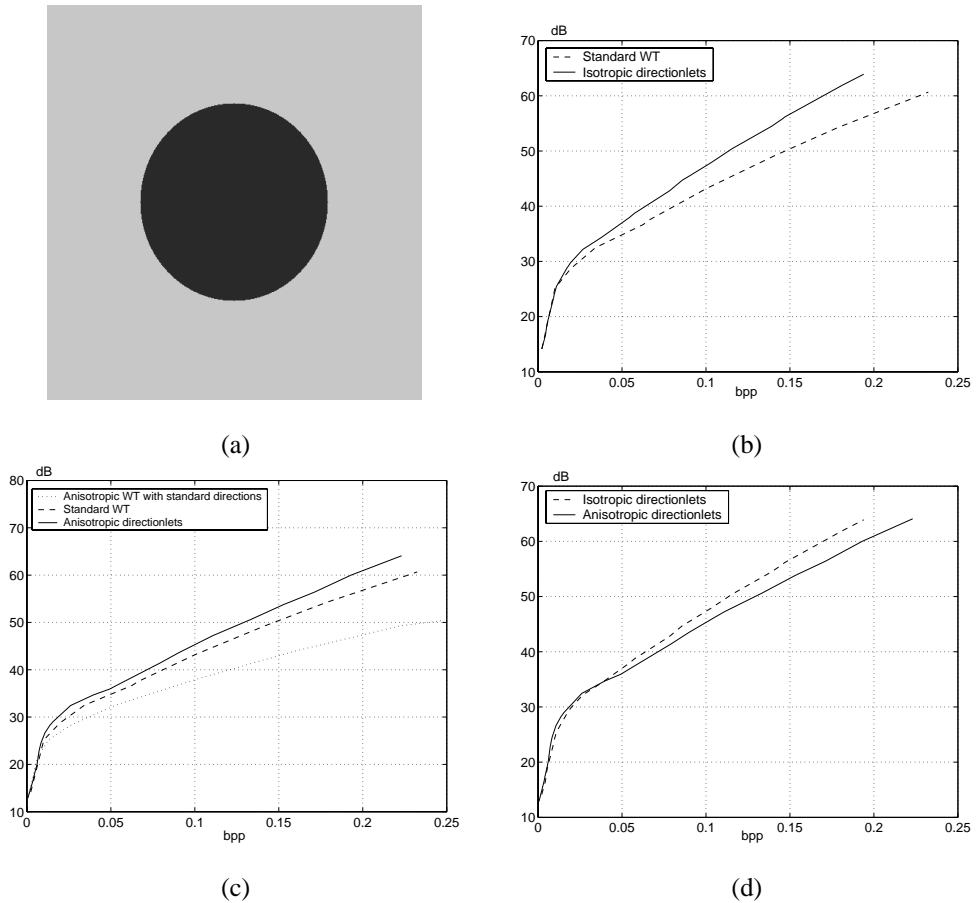


Figure 5.13: Synthetic image with a circle and compression efficiency of various schemes. (a) A bilevel image with a circle. (b) A comparison between the compression methods using the standard WT and the M-DIR isotropic transforms with the optimal spatial segmentation and the optimal choice of directions from the set $\{0^\circ, 90^\circ, 45^\circ, -45^\circ\}$. The optimized method outperforms the method that uses the standard WT. (c) A comparison between the compression methods using the standard WT, the anisotropic S-AWT($M_\Lambda, 2, 1$) with the standard directions, and the same anisotropic transform with the optimal segmentation and choice of directions. The performance of the optimal anisotropic transform is the best. (d) The best anisotropic transform from (c) outperforms the best isotropic transform from (b) at very low bitrates. However, the isotropic transform is better at higher bitrates as a consequence of a stronger border effect in the case of the anisotropic transform.

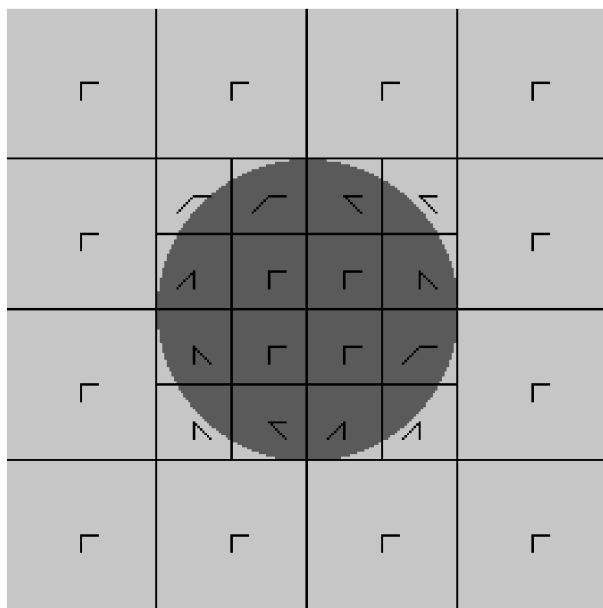


Figure 5.14: The optimal segmentation and the choice of transform and alignment directions for the simple image shown in Figure 5.13(a) in the case of the optimized compression method using the isotropic transform.

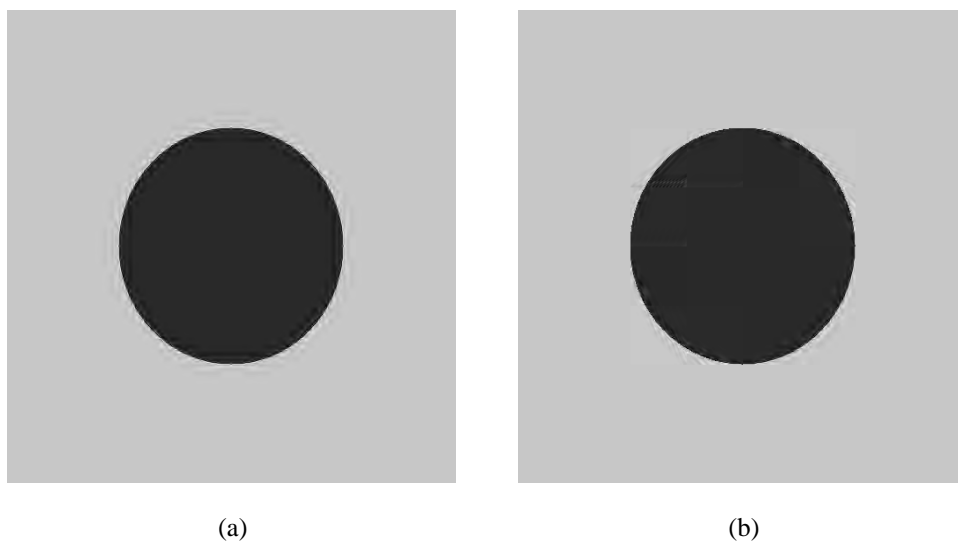


Figure 5.15: Two reconstructed images of the circle shown in Figure 5.13(a) using (a) the standard WT and (b) the isotropic transform with the optimal segmentation and choice of directions at the bitrate $R = 0.0378\text{bpp}$. The resulting distortions are: (a) $D = 32.79\text{dB}$ and (b) $D = 34.35\text{dB}$.

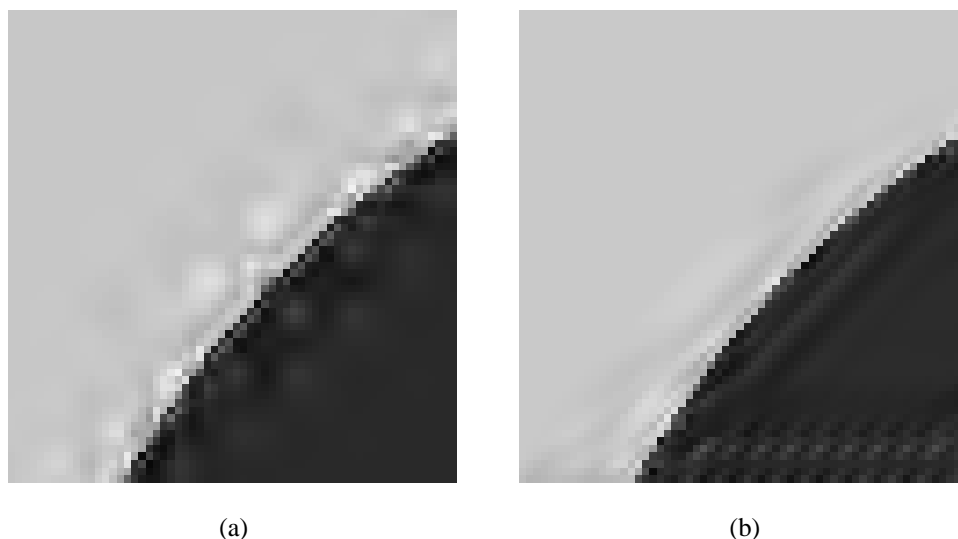


Figure 5.16: The two reconstructed images shown in Figure 5.15 are magnified and the arc of the circle is shown for (a) the standard WT and (b) the isotropic transform with the optimal segmentation and choice of directions. The ringing around the arc is more significant in the case of the standard WT.

Finally, we show the results of the optimized compression algorithm applied on natural images, in particular on Lena, Barbara and Goldhill. These images and the corresponding results are shown in Figures 5.20-5.22 for the standard WT and the optimized method using the anisotropic transforms. The results of the standard method coincide well with the results provided in [81]. Notice that the gain obtained using the optimized method over the standard method is much smaller than in the case of synthetic images. In particular, the new method significantly outperforms the standard at a high compression rate, that is, for the low bitrate. The results of compression of the image Lena are shown in Figure 5.23, where the comparison from Figure 5.20(b) is zoomed in and the two reconstructed images are shown for the bitrate $R = 0.0395\text{bpp}$. The PSNR factor of the second reconstruction is better than the same in the case of the standard WT.

The compression efficiency of the optimized method applied on natural images is limited by the border effect produced by wrapping the filters around the borders of the image. In general, the extension pixels that are artificially generated are not smooth with respect to the order of vanishing moments of the HP filters and, thus, the HP filtering across them produces large magnitude transform coefficients. Therefore, a region around the borders of image that contains these coefficients requires many bits for encoding the significance map and quantization.

In order to improve the efficiency of the zerotree-based compression methods the number of decomposition levels in the transform is required to be large so that the prediction of locations of low-magnitude coefficients across scales is more efficient and the number of coding symbols is decreased. On the other hand, in order to reduce the impact of the border effect, the 1-D filters used in the transform are required to be shorter so that the area of image

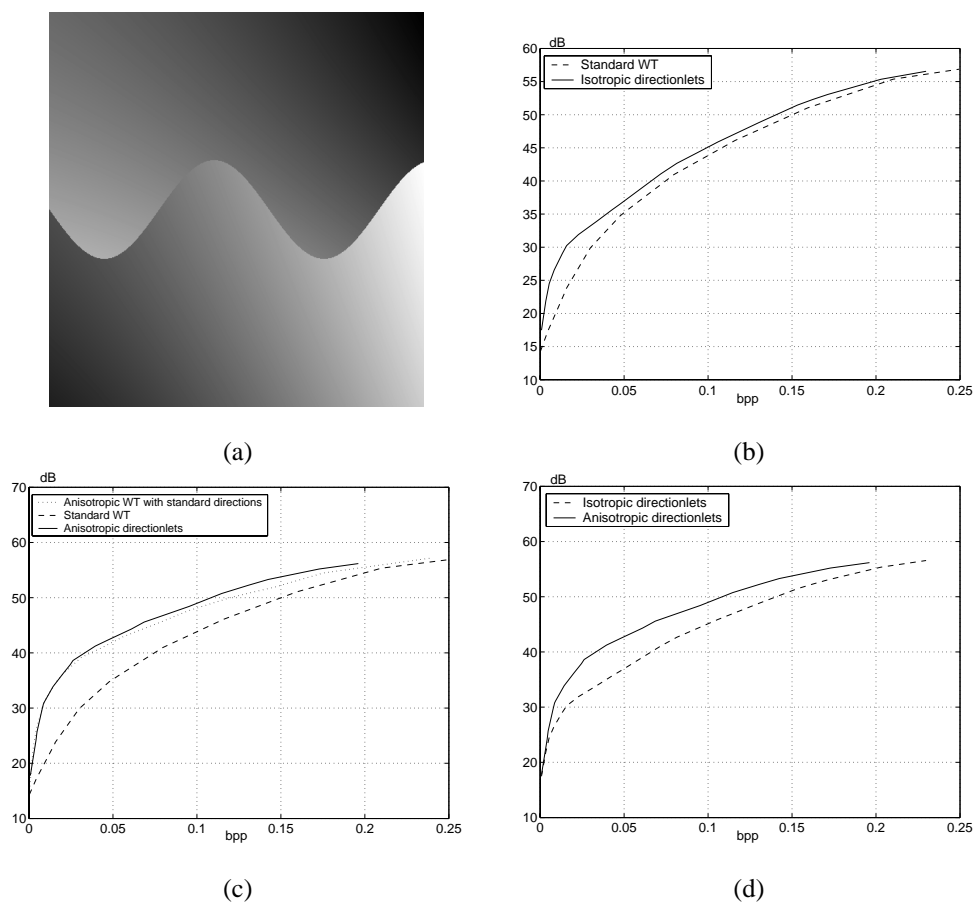


Figure 5.17: An image from the class C^2/C^2 and the compression efficiency of various schemes. (a) An image from the class C^2/C^2 . (b) A comparison between the compression methods using the standard WT and the M-DIR isotropic transforms with the optimal spatial segmentation and the optimal choice of directions from the set $\{0^\circ, 90^\circ, 45^\circ, -45^\circ\}$. The optimized method outperforms the standard method constantly for a wide range of bitrates. (c) A comparison between the compression methods using the standard WT, the anisotropic S-AWT($M_\Lambda, 2, 1$) with the standard directions, and the same anisotropic transform with the optimal segmentation and choice of directions. The optimized method using the anisotropic transform is significantly better than the method using the standard transform. (d) The performance of the anisotropic transform is better than that of the isotropic transform. The anisotropic basis functions are better adapted to the anisotropic object in this image providing a sparser representation and better compression efficiency.

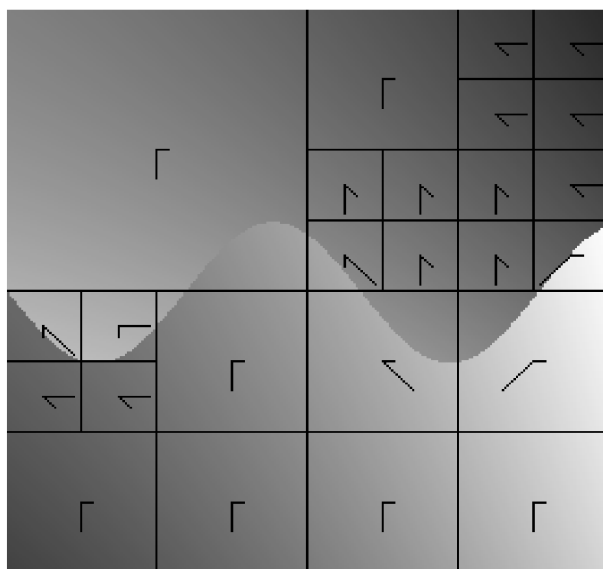


Figure 5.18: The optimal segmentation and the choice of transform and alignment directions for the image shown in Figure 5.17(a) in the case of the S-AWT($M_{\Lambda,2,1}$). The longer lines determine the transform direction in each segment, whereas the shorter lines represent the alignment directions.

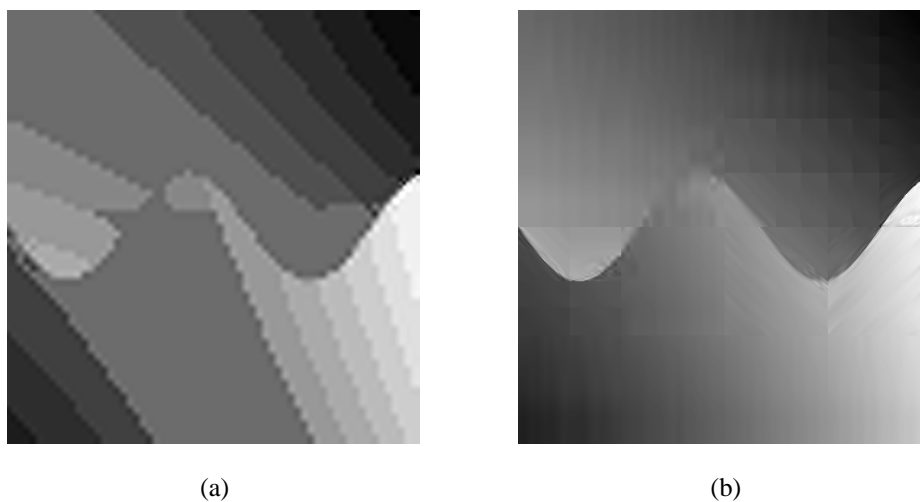


Figure 5.19: Two reconstructed images of the image shown in Figure 5.17(a) using (a) the standard WT and (b) the anisotropic transform S-AWT($M_{\Lambda,2,1}$) with the optimal segmentation and choice of directions at the bitrate $R = 0.0263bpp$. The anisotropic transform substantially outperforms the standard WT in both the numerical result and visual quality of the reconstructions. The resulting distortions are: (a) $D = 26.33dB$ and (b) $D = 38.62dB$.

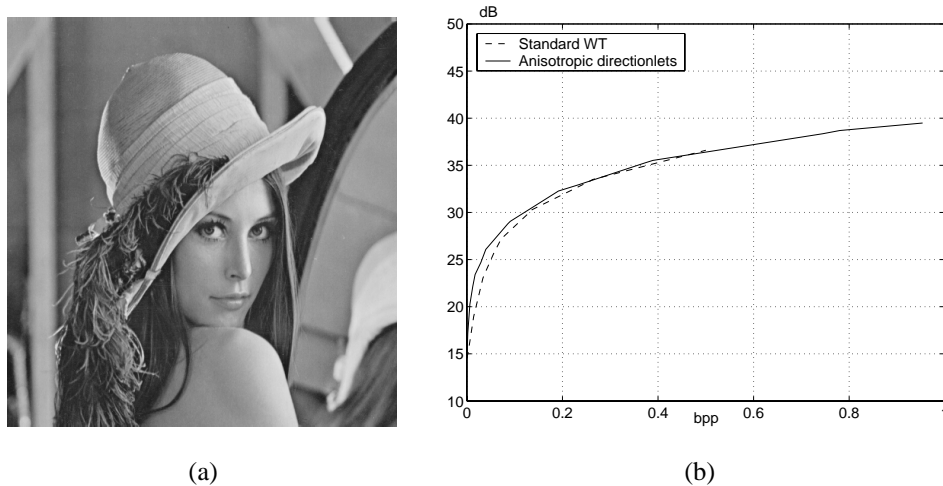


Figure 5.20: A natural image and the compression efficiency. (a) The standard test image Lena. (b) The compression results of Lena using the standard WT in the zerotree-based compression method and the anisotropic transforms in the optimized zerotree-based method.

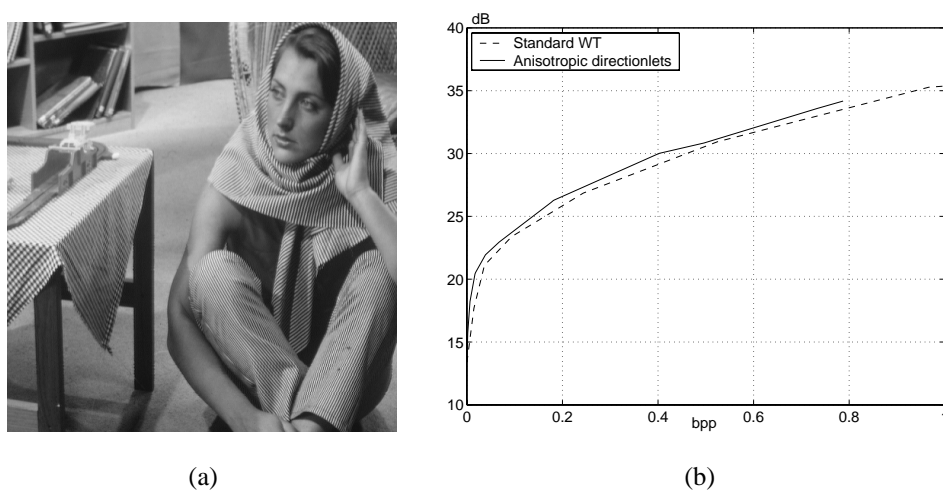


Figure 5.21: A natural image and the compression efficiency. (a) The standard test image Barbara. (b) The compression results of Barbara using the standard WT in the zerotree-based compression method and the anisotropic transforms in the optimized zerotree-based method.

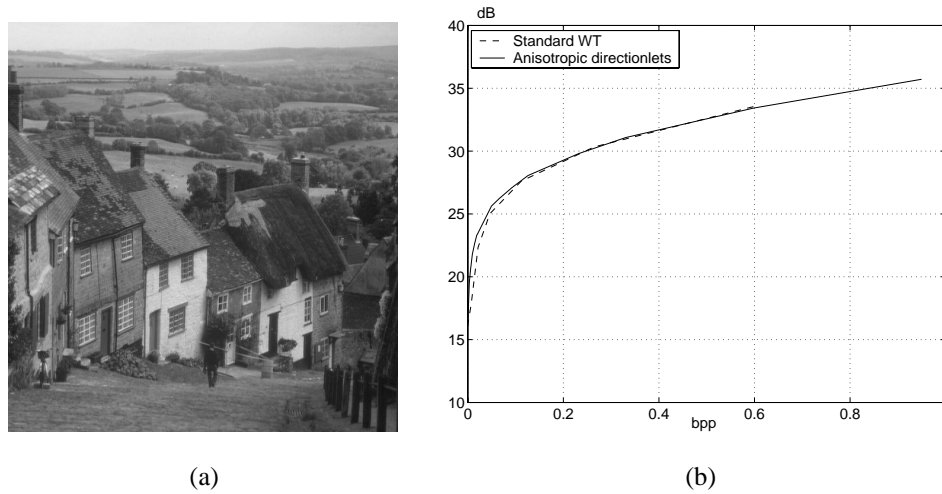


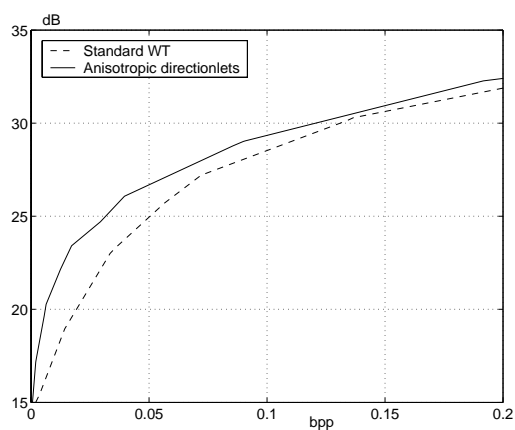
Figure 5.22: A natural image and the compression efficiency. (a) The standard test image Goldhill. (b) The compression results of Goldhill using the standard WT in the zerotree-based compression method and the anisotropic transforms in the optimized zerotree-based method.

affected by the border effect is smaller. Recall that the support of the equivalent 1-D filters at coarser scales is longer. Therefore, these two requirements are in contradiction, that is, the more decomposition levels are applied in the transform, the wider the area affected by the border effect is. Furthermore, even though spatial segmentation contributes to sparsity of the transform allowing for adaptation of transform and alignment directions in each segment, it enlarges the total size of the region affected by the border effect in the image because the extension pixels are generated for each segment independently.

Notice that the large magnitude coefficients generated by filtering across the extension pixels are aligned with the borders of segments, which are, in turn, horizontal and vertical. This is the reason why, in the case of natural images, the transform along these two standard directions provides often the best compression results in the sense of Lagrangian optimization. As a consequence, spatial segmentation does not reduce the cost of encoding and, therefore, the optimization compression method reduces to the standard zerotree-based algorithm. However, in the case of synthetic images, the optimal method does not coincide with the standard algorithm because it is much easier to have extension pixels that do not generate a significant border effect.

Several extension methods that perform minimization of the border effect in the case of both 1-D and 2-D signals have been previously proposed. For instance, in boundary wavelets, the filters that are wrapped around the border of signal are modified so that their support is completely inside the signal, while orthogonality is preserved [42; 43]. In [63; 94], the extension pixels are computed so that the square norm of the distance to the first sample in the original support of the signal is minimized. A more general analysis of linear smooth extensions yielding non-expansive transforms is given in [29].

In our work, we use the symmetric extension that has been shown to outperform the periodic extension in terms of the energy in the HP subbands, while retaining critical sampling



(a)



(b)



(c)

Figure 5.23: The results of compression of the image Lena at low bitrate. (a) Magnified comparison of the two methods shown in Figure 5.20(b). (b) Reconstruction of the image compressed using the standard method at $R = 0.0395\text{bpp}$ and $D = 23.51\text{dB}$. (c) Reconstruction of the image compressed by the optimized method at the same bitrate $R = 0.0395\text{bpp}$ and $D = 26.07\text{dB}$.

in the case of biorthogonal symmetric filters [8]. However, the symmetric extension is not efficient enough in compression of natural images because the impact of border effects is still too large.

Even though the boundary wavelets provide an elegant way to prevent the border effect in the case of the 1-D or the standard 2-D WT, they are not applicable in the case of the lattice-based transforms. Recall from Chapter 3 and Figure 3.4 that the lengths of lines and the subsampling shifts can vary across the image domain depending on the transform and alignment directions. However, the construction of the 1-D boundary wavelets is constrained only on even shifts of subsampling, which makes the method too complex for the use in practice.

On the other hand, adding extension pixels causes oversampling in general (with an exception in the case of symmetric or periodic extension). Notice that minimization of the impact of border effects in one scale of the transform is equivalent to minimization of the energy of the HP response that corresponds to filtering across the extension pixels. However, even though the energy of this HP border response can be made arbitrarily small in the case of the biorthogonal "9-7" HP filter, the corresponding LP response blows up. That leads to a very inefficient compression in the case of the iterated transform. Furthermore, a constrained minimization of the energy of the HP output with upper-bounded energy in the LP subband is shown to be too complex for implementation in practice. For that reason and the reason of critical sampling, we retain the symmetric extension in the applications of the compression method.

Recall that there exist newer and more efficient compression algorithms based on the WT than the one proposed in [81]. However, in these algorithms, similar structures of wavelet coefficients are built. These structures can be modified in a similar manner as proposed in this chapter using anisotropic grouping of coefficients. For that reason we do not compare our results to the compression results given in, for instance, [79; 113; 114].

5.6 Conclusions

In this chapter, we have shown that directionlets with spatial segmentation outperform the standard WT in terms of asymptotic behavior in NLA and compression of C^2/C^2 images. Furthermore, the performance of the zerotree-based compression algorithm using directionlets applied on C^2/C^2 images is substantially better when compared to the standard method. However, the success of compression using directionlets is limited in the case of natural images because of the border effect. The further improvements in this case are left for future work.

5.A Proofs

5.A.1 Proof of Equation 5.1

Assume that, given a frame $\mathbf{F} \in \mathbb{R}^{m \times n}$, the vector $\mathbf{y} \in \mathbb{R}^m$ is defined as $\mathbf{y} = \mathbf{F}\mathbf{x}$ for any $\mathbf{x} \in \mathbb{R}^n$. Here $m \geq n$. Recall that the pseudo-inverse transform is given by $\mathbf{x} = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T \cdot \mathbf{y}$ [56]. Recall also that if the frame \mathbf{F} is tight, then $\|\mathbf{y}\|_2^2 = A \|\mathbf{x}\|_2^2$, where A is called the *frame bound*. This is equivalent to

$$\mathbf{F}^T \cdot \mathbf{F} = A \mathbf{I}_n, \quad (5.5)$$

where \mathbf{I}_n is the $n \times n$ identity matrix. In that case, the inverse transform is simplified and it is given by $\mathbf{x} = A^{-1} \mathbf{F}^T \cdot \mathbf{y}$.

Now, assume that a non-linear operator (e.g. NLA, thresholding, etc.) $T : \mathbb{R}^m \rightarrow \mathbb{R}^m$ is applied on \mathbf{y} yielding $\hat{\mathbf{y}}$, that is, $\hat{\mathbf{y}} = T(\mathbf{y})$. Thus, $\hat{\mathbf{x}} = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T \cdot \hat{\mathbf{y}}$.

The MSE in the original domain is defined as $\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2$ and, similarly, the MSE in the transform domain is given by $\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$. Assuming that the frame \mathbf{F} is tight we can write

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 = \left\| \frac{1}{A} \mathbf{F}^T \cdot (\mathbf{y} - \hat{\mathbf{y}}) \right\|_2^2 \leq \frac{1}{A^2} \|\mathbf{F}^T\|_2^2 \cdot \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2,$$

where equality holds when \mathbf{F} is orthogonal.

From [59], we have that $\|\mathbf{F}^T\|_2^2 = \|\mathbf{F}\|_2^2 = A$. Hence, the MSE in the original and transform domains are related as

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 \leq \frac{1}{A} \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2.$$

5.A.2 Proof of Theorem 5.1

Recall first that a C^2 curve can be locally represented by the Taylor series expansion, that is, by a quadratic polynomial

$$y(x) = ax^2 + bx + c, \quad (5.6)$$

where a and b are related to the second and first derivative of the curve (curvature and linear component), respectively. Without loss of generality, we assume that the C^2 discontinuity curve is *Horizon* [31] on the unit square $[0, 1]^2$.

Since the smooth regions of the function $f(x_1, x_2)$ are C^2 , assume that the 1-D filters used in the S-AWT are orthogonal and have at least two vanishing moments. Let the transform be applied along the class of straight lines defined by

$$\{y(x) = rx + d : d \in \mathbb{R}\}. \quad (5.7)$$

Here, the slope r determines the transform direction, whereas the alignment direction is vertical. Equalizing (5.6) and (5.7) we can write

$$d(x) = ax^2 + (b - r)x + c.$$

The transform coefficients of the S-AWT that intersect the discontinuity curve are called *E-type coefficients*. The number of the E-type coefficients at the scale j is given by $N_e^{(0)}(j) = O(2^{n_2 j} \Delta_d)$. Here, n_2 is the number of transforms applied along the vertical direction, $\Delta_d = \max_{0 \leq x \leq 1} d(x) - \min_{0 \leq x \leq 1} d(x)$ is the width of the strip along the transform direction that contains the curve (see Figure 5.24), and zero in the superscript of $N_e^{(0)}(j)$ denotes that no segmentation has been applied yet. The transform direction with the slope

$$r = a + b \quad (5.8)$$

minimizes the width Δ_d (and, thereof, $N_e^{(0)}(j)$) on the unit square. In that case the number of the E-type coefficients is given by

$$N_e^{(0)}(j) = O\left(\frac{a}{4} 2^{n_2 j}\right).$$

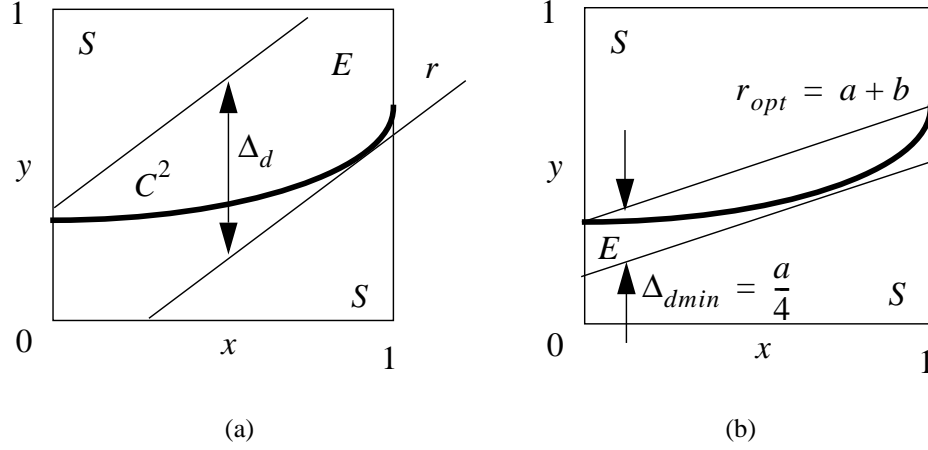


Figure 5.24: The 2-D function $f(x_1, x_2)$ is C^2 smooth on the unit square away from a C^2 discontinuity curve. The curve can be locally approximated by a quadratic polynomial $y(x) = ax^2 + bx + c$. The E-type transform coefficients intersect the curve and have a slower decay of magnitudes across scales than the S-type coefficients, which correspond to the smooth regions. (a) The S-AWT produces the E-type coefficients within the strip along the slope r . (b) The width of the strip Δ_d is minimized for $r = a + b$.

Notice that an increment in the scale index j is equivalent to a step to a finer scale.

The transform coefficients of the S-AWT, which do not intersect the discontinuity curve are called *S-type coefficients*. The number of the S-type coefficients depends on the number of transforms n_1 and n_2 at a scale along the transform and vertical directions, respectively, as

$$N_s^{(0)}(j) = 2^{(n_1+n_2)j} - N_e^{(0)}(j) = O\left(2^{(n_1+n_2)j} - \frac{a}{4}2^{n_2j}\right).$$

An anisotropic spatial segmentation is applied on the unit square. It partitions the unit square into vertical strips using the dyadic rule, that is, there are 2^s vertical strips at the s th level of segmentation, where the width of each is 2^{-s} (Figure 5.25). The optimal transform direction, according to (5.8), is chosen for each segment independently. Since each segment is rescaled again to the unit square, the number of the E-type transform coefficients in a segment is reduced and is given by

$$O\left(\frac{a}{4}2^{n_2j} \cdot 2^{-2s}\right).$$

The total number of the E-type coefficients is given by the sum across all the segments, that is,

$$N_e(j, s) = \sum_{k=0}^{2^s-1} O\left(\frac{a}{4}2^{n_2j-2s}\right) = O\left(\frac{a}{4}2^{n_2j-s}\right). \quad (5.9)$$

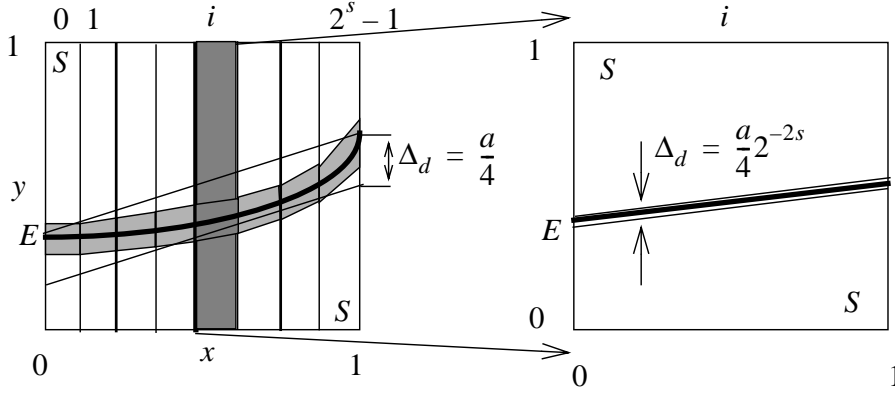


Figure 5.25: Anisotropic segmentation partitions the unit square into 2^s equally wide vertical strips. After rescaling, the curvature parameter a (related to the second derivative of the C^2 curve) is reduced in each segment by the factor 2^{2s} . Since there are 2^s segments that intersect the discontinuity, the total number of the E-type transform coefficients is reduced by 2^s . At the same time, the total number of the S-type coefficients is increased by the same factor.

Similarly, the total number of the S-type coefficients is given by

$$N_s(j, s) = \sum_{k=0}^{2^s-1} O\left(2^{(n_1+n_2)j} - \frac{a}{4}2^{n_2j-2s}\right) = O\left(2^{(n_1+n_2)j+s} - \frac{a}{4}2^{n_2j-s}\right). \quad (5.10)$$

Notice that the exact number of the two types of coefficients given by (5.9) and (5.10) depends on the length of the 1-D filters used in the transform. However, the dependence is only up to a constant and, thus, the order of growth of these numbers across scales remains the same.

The magnitudes $|w_e(j)|$ of the E-type coefficients decay across scales as

$$O\left(2^{-(n_1+n_2)j/2}\right).$$

The S-type coefficients correspond to the smooth regions of the function $f(x_1, x_2)$ and their magnitudes $|w_s(j)|$ are upper bounded by $O(2^{-n_3j/2})$. Notice that, since the 1-D HP filters have vanishing moments, the decay of the magnitudes of the S-type coefficients is faster than the one of the E-type coefficients, that is, $n_3 > n_1 + n_2$.

We estimate n_3 considering that the applied 1-D wavelets have at least two vanishing moments. It is shown in [56] that, the decay of the magnitudes $|w_s(j)|$ in a smooth region after two consecutive transforms with alternated transform directions is 2^{-3} . Therefore, the decay rate n_3 is given by

$$n_3 = 6 \cdot \min(n_1, n_2) + |n_2 - n_1| = \begin{cases} n_1 + 5n_2, & n_1 \geq n_2 \\ 5n_1 + n_2, & n_1 \leq n_2 \end{cases}. \quad (5.11)$$

To approximate the function $f(x_1, x_2)$, we keep all the coefficients with the magnitudes larger than or equal to the threshold 2^{-m} , where $m \geq 0$, and discard (set to zero) the others. The retained coefficients can be divided into two groups:

- (1) The E-type coefficients at the scales $0 \leq j \leq 2m/(n_1 + n_2)$,
- (2) The S-type coefficients at the scales $0 \leq j \leq 2m/n_3$.

From (5.9), (5.10) and decays of the magnitudes across scales, we compute the order of the total number of retained coefficients $N(m, s)$ and the MSE. The number $N(m, s)$ is the sum of the retained E and S-type coefficients:

$$\begin{aligned} N(m, s) &= \sum_{j=0}^{2m/(n_1+n_2)} N_e(j, s) + \sum_{j=0}^{2m/n_3} N_s(j, s) \\ &= O\left(2^{\frac{2n_2}{n_1+n_2}m-s}\right) + O\left(2^{\frac{2(n_1+n_2)}{n_3}m+s}\right). \end{aligned} \quad (5.12)$$

The MSE is given by

$$\begin{aligned} \text{MSE}(m, s) &= \sum_{j=2m/(n_1+n_2)+1}^{+\infty} N_e(j, s)|w_e(j)|^2 + \sum_{j=2m/n_3+1}^{+\infty} N_s(j, s)|w_s(j)|^2 \\ &= O\left(2^{-\frac{2n_1}{n_1+n_2}m-s}\right) + O\left(2^{-\frac{2(n_3-n_1-n_2)}{n_3}m+s}\right). \end{aligned} \quad (5.13)$$

Assuming that the number of segmentation levels depends on the exponent m of the threshold as $s = \eta m$, where the *segmentation rate* $\eta \geq 0$, we distinguish the two cases, as follows:

- (1) The terms in (5.12) and (5.13) produced by the E-type coefficients dominate, in which case we have

$$\eta \leq \eta^* = \frac{n_2}{n_1 + n_2} - \frac{n_1 + n_2}{n_3} = \frac{1}{\rho + 1} - \frac{\rho + 1}{\rho + 5},$$

where $\rho = n_1/n_2 \geq 1$. Then the MSE decays as

$$\text{MSE} = O(N^{-e_1}), \text{ where } e_1 = \frac{2n_1 + \eta(n_1 + n_2)}{2n_2 - \eta(n_1 + n_2)} = \frac{2\rho + \eta(\rho + 1)}{2 - \eta(\rho + 1)}.$$

- (2) The terms in (5.12) and (5.13) produced by the S-type coefficients dominate, that is, $\eta \geq \eta^*$ and

$$\text{MSE} = O(N^{-e_2}), \text{ where } e_2 = \frac{2(n_3 - n_1 - n_2) - \eta n_3}{2(n_1 + n_2) + \eta n_3} = \frac{8 - \eta(\rho + 5)}{2(\rho + 1) + \eta(\rho + 5)}.$$

Plugging (5.11) in the relations above and knowing that the segmentation rate η is a non-negative value, we obtain the maximal decay rate $\text{MSE} = O(N^{-\alpha})$, with $\alpha = (\sqrt{17} - 1)/2 \approx 1.562$. The optimal rate is attained for the anisotropy ratio $\rho^* = n_1/n_2 = \alpha \approx 1.562$ and the segmentation rate $\eta^* = 0$.

Notice that the analysis above is based on two assumptions: (a) the optimal transform direction given by (5.8) is chosen and (b) the C^2 curve is globally represented by a quadratic polynomial given by (5.6). Here, we address these two assumptions showing that they do not constrain severely the approximation rate.

- (a) Assume that the transform direction is given by the suboptimal slope $r = a + b + \epsilon$, where $|r| \leq 1$. Then it can be shown that $\Delta_d = a/4 + |\epsilon|/2 + \epsilon^2/4a = O(a)$ for $|\epsilon| \leq a$ and $\Delta_d = |\epsilon| + o(\epsilon)$ for $|\epsilon| > a$. Furthermore, assume that ϵ decays exponentially with the number of segmentation steps, that is, $\epsilon \sim 2^{-\beta s}$, where $\beta > 0$. If $\beta < 2$, then the expression of $N_e(j, s)$ given by (5.9) becomes $O(2^{n_2 j - (\beta-1)s})$ and the optimal segmentation rate η^* is multiplied by the factor $2/\beta$. In that case the exponent e_1 is given by

$$e_1 = \frac{2\rho + (\beta - 1)\eta(\rho + 1)}{2 - (\beta - 1)\eta(\rho + 1)},$$

whereas the exponent e_2 is unchanged. However, even though some of these parameters are changed, the optimal approximation rate remains the same, that is, $\text{MSE} = O(N^{-\alpha})$ if $\rho^* = \alpha$ and $\eta^* = 0$. On the other hand, now the required number of transform directions is finite and behaves as $1/|\epsilon| \sim 2^{\beta s}$.

- (b) The analysis that leads to the approximation rate holds only for the case when the slope of the tangent direction (or, equivalently, the first derivative) of the C^2 curve is in the interval $[-1, 1]$. However, the first derivative of a general C^2 curve is not constrained on that interval and, therefore, the optimal approximation rate cannot be achieved in the same way as in the case of a quadratic polynomial. In order to be able to achieve the same rate we need to introduce an initial number of segmentation steps prior to the iteration. Recall that one step of anisotropic segmentation attenuates twice the first derivative of the C^2 curve.⁶ Thus, it suffices to apply enough segmentation steps so that the maximal magnitude of the first derivative is less than or equal to 1. Then, the iterated segmentation and transform are continued on each of these initial segments and this construction results in the same optimal approximation rate. Notice that the necessity for reducing the magnitude of the first derivative below 1 is caused by the assumption that the C^2 curve is Horizon. However, if this assumption is not satisfied, then an appropriate combination of initial segmentation steps and transposition of the axes can rescale the curve so that each segment of the curve is Horizon. Therefore, the optimal approximation rate can be achieved in the case of a general C^2 curve.

For the compression application, the retained coefficients have to be indexed and quantized. For a given MSE (or *distortion*) each of these operations carries a cost in terms of the required bits.

The N retained S-AWT coefficients within a spatial segment can be organized in an embedded tree-structure, similar to the structures produced by the standard WT and exploited in the other compression algorithms (zero-trees [81], SPIHT [79], SFQ [113; 114]). The main difference between the tree-structures of the standard WT and S-AWT is in the number of descendants of each transform coefficient. While this number is fixed in the standard WT, it depends on the number of transform steps applied at each scale in the S-AWT. However, the S-AWT tree-structure allows also for indexing the retained coefficients using approximately 1 bit per transform coefficient.

A variable length coding scheme allocates l bits to encode coefficients with magnitudes in the interval $[2^{-m}2^{l-1}, 2^{-m}2^l)$. Thus, using (5.12) and the optimal choice for n_1, n_2, n_3 ,

⁶One step of the anisotropic segmentation is equivalent to stretching the abscissa by the factor 2 and, therefore, the equivalent first derivative of the curve is also attenuated by 2.

and η , the total number of encoding bits R is given by:

$$\begin{aligned} R(m) &= N(m, 0) + \sum_{l=1}^{\infty} N(m-l, 0) \\ &= O(2^{\frac{\alpha}{2}m}) + \sum_{l=1}^{\infty} 2^{\frac{\alpha}{2}(m-l)} = O(2^{\frac{\alpha}{2}m}). \end{aligned} \quad (5.14)$$

The distortion D consists of two components: (a) the MSE resulting from the truncation of small coefficients in the approximation given by (5.13), and (b) distortion caused by the quantization of the retained coefficients. The second component is given by $N(m, 0) \cdot 2^{-2m}$ and, thus, the total distortion is

$$D(m) = \text{MSE}(m, 0) + N(m, 0) \cdot 2^{-2m} = O\left(2^{-\frac{\alpha}{2}m}\right). \quad (5.15)$$

The R-D behavior follows from (5.14) and (5.15) and it is given by

$$D(R) = O(R^{-\alpha}).$$

Chapter 6

Denoising of Images using Directionlets

6.1 Introduction

The goal of image denoising is to reconstruct an original image from an observation corrupted by noise. The statistical model for both the noise and original image is the key to achieving a good quality of reconstruction.

However, an image is a complex signal and, thus, finding a good statistical model is a non-trivial task. In order to simplify the statistical modeling, two assumptions are usually made:

- (1) Pixels obey a local Markov model, that is, the probability structure is defined locally, on a set of neighbor pixels and it is independent of the pixels out of the neighborhood.
- (2) The distribution of values in a neighborhood is space invariant, that is, it does not depend on the absolute spatial location of the pixel.

These two assumptions result in a Markov random field.

Statistical modeling of images is substantially simplified by transforming the image into a new representation. Nowadays, in image denoising, the implemented transforms perform image decomposition into a multiscale and multi-directional (M-DIR) representation. These transforms share some properties with neural responses in the primary visual cortex of mammals, which are very important in efficient representation of visually relevant features of images [109].

In the recent literature, the marginal distributions among the transform coefficients are often modeled as Gaussian or generalized Gaussian [2]. However, the transform coefficients of natural images have marginal distributions with heavier tails than the generalized Gaussian and, thus, the generalized Gaussian does not describe efficiently these local statistics. A better modeling has been achieved by *hidden Markov models*, where a simple parametric model is influenced by a set of hidden random variables. Now, even the generalized Gaussian random variable can be used as the parameter, but the equivalent marginal distributions have heavier tails and model well the distributions of the transform coefficients [22; 73]. Here, we use a version of the hidden Markov model called *Gaussian scale mixtures* (GSM) [1].

In this chapter, we review the definition and properties of the GSM estimation model when applied along with a multiscale M-DIR transform representation of images, as explained in [68]. Then, we describe the denoising method that uses the GSM model and the *undecimated* version of directionlets. Finally, we show the experimental results obtained using the denoising method applied on different natural images.

6.2 Gaussian Scale Mixtures Estimation Model

Gaussian scale mixtures have been successfully used to describe statistical properties of wavelet coefficients [105; 106]. This model represents well the heavy tailed marginal distribution of the coefficients in the wavelet domain and the strong correlation between the amplitudes of neighbor coefficients.

Here, we review the single-step least square optimal Bayesian estimator proposed in [68]. The estimator computes the optimal estimation of the reference coefficient from the neighbor coefficients in the transform domain. The neighborhood of a referent coefficient consists of the coefficients in the neighbor spatial locations at the same scale and the parent coefficient, that is, the coefficient in the same spatial location, but at the next coarser scale. A typical neighborhood used in [68] is illustrated in Figure 6.1.

6.2.1 Image Model

The transform coefficients of an image are modeled as GSM vectors, where, for each coefficient, the coefficients from the corresponding neighborhoods form the vector, as shown in Figure 6.1. The coefficients at the coarsest scale are an exception because the corresponding neighborhoods do not contain the parents.

In general, a random vector \mathbf{x} is a GSM if and only if it can be expressed as the product of a zero-mean Gaussian vector \mathbf{u} and an independent positive scalar random variable \sqrt{z} [1], that is,

$$\mathbf{x} = \sqrt{z}\mathbf{u}, \quad (6.1)$$

where the equality holds in distribution. The variable z is called the *multiplier*. The vector \mathbf{x} is, thus, an infinite mixture of Gaussian vectors, whose density is determined by the covariance matrix $\mathbf{C}_{\mathbf{u}}$ of the vector \mathbf{u} and the mixing density $p_z(z)$ as

$$p_{\mathbf{x}}(\mathbf{x}) = \int p(\mathbf{x}|z)p_z(z)dz = \int \frac{\exp\left(\frac{-\mathbf{x}^T(z\mathbf{C}_{\mathbf{u}})^{-1}\mathbf{x}}{2}\right)}{(2\pi)^{N/2}|z\mathbf{C}_{\mathbf{u}}|^{1/2}}p_z(z)dz, \quad (6.2)$$

where N is the dimensionality of \mathbf{x} and \mathbf{u} . Without loss of generality, one can assume $E\{z\} = 1$, which implies equality of the covariance matrices $\mathbf{C}_{\mathbf{x}}$ and $\mathbf{C}_{\mathbf{u}}$.

The GSM random vectors include both the generalized Gaussian and the α -stable family of random variables. The probability density of GSM is symmetric and zero-mean, and has heavier tails than the Gaussian marginal densities. A key property of GSM is that the density of $p_{\mathbf{x}}(\mathbf{x}|z)$ is Gaussian as well as the normalized vector \mathbf{x}/\sqrt{z} .

The probability density $p_z(z)$ of the multiplier has to be specified *a priori*. Among several choices proposed in [68] we use the *non-informative prior* (see also [5]) because it shows the best performance in the denoising experiments.

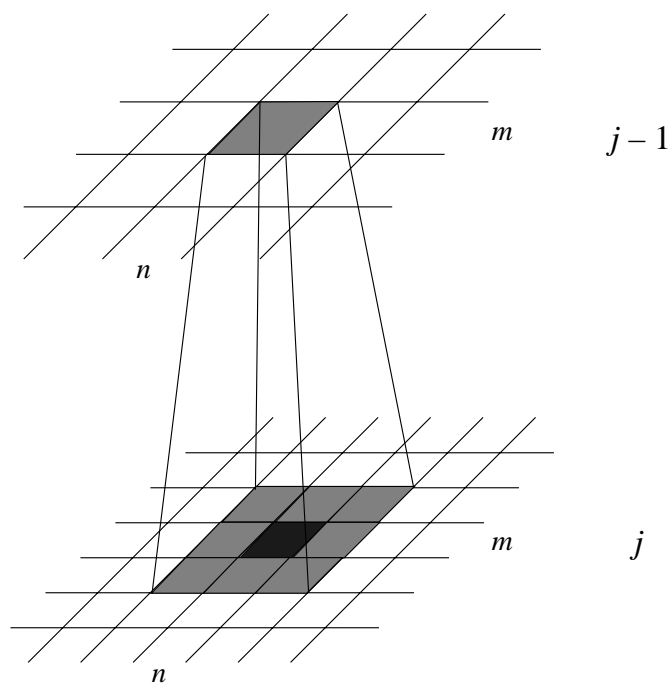


Figure 6.1: Neighborhood around a referent coefficient consists of the coefficients in the neighbor spatial locations at the same scale j and the coefficient in the same spatial location as the referent coefficient, but at the next coarser scale $j - 1$, which is called parent.

The non-informative prior is convenient for the application in denoising of general images because it does not require the fitting of any parameter to the noisy observation. The most common solution is known as *Jeffrey's prior* and it has also been used in [39] to define marginal priors for image denoising. In this case, the probability density $p_z(z)$ is estimated as

$$p_z(z) \propto \sqrt{I(z)}, \quad (6.3)$$

where $I(z)$ is the Fisher information matrix given by

$$I(z) = E \left\{ -\frac{\partial^2 \log p(\mathbf{x}|z)}{\partial z^2} \right\}.$$

For the GSM model, it follows from (6.1) and (6.2) that

$$\begin{aligned} -\frac{\partial^2 \log p(\mathbf{x}|z)}{\partial z^2} &= \frac{\partial^2}{\partial z^2} \left[\frac{1}{2} \left(N \log(z) + \log |\mathbf{C}_u| + \frac{\mathbf{x}^T \mathbf{C}_u^{-1} \mathbf{x}}{z} \right) \right] \\ &= \frac{N}{2z^2} + \frac{\mathbf{x}^T \mathbf{C}_u^{-1} \mathbf{x}}{2z^3}. \end{aligned} \quad (6.4)$$

Computing (6.3) and using the fact that $E\{\mathbf{x}^T \mathbf{C}_u^{-1} \mathbf{x}\} = z$, we obtain Jeffrey's prior

$$p_z(z) \propto \frac{1}{z}, \quad (6.5)$$

which corresponds to a constant function on $\log(z)$. Notice that, even though this is an improper probability density, it is common to ignore that fact as long as it does not create computational problems. In the implementation, we set the prior to zero in the interval $[0, z_{min}]$, where z_{min} is a small positive constant.

6.2.2 Noise Model

The noise added to the pixels in the original image domain is assumed to be *iid additive white Gaussian*. The standard deviation σ of the noise is either known a priori or can be estimated using the robust estimator [14; 15; 20; 30; 32]

$$\hat{\sigma} = \frac{\text{Median}(|y(n, m)|)}{0.6745}, \quad (6.6)$$

where the coefficients $y(n, m)$ are taken from the finest subband of the wavelet transform.

In the transform domain, the corresponding coefficients produced by the transform of the noise are correlated, since a non-orthogonal transform is used. We assume a local Markov model, that is, the referent transform coefficient is correlated only with the transform coefficients in the closest spatial neighborhood at the referent and the next coarser scale, as illustrated in Figure 6.1.

The covariance of the corresponding transform coefficients in the neighborhood is estimated by decomposing a delta function $\sigma \sqrt{NM} \cdot \delta(n, m)$ into the transform subbands, where (N, M) are the image dimensions. This signal has the same power spectrum as the noise, but it is free from random fluctuations. For each subband, elements of the covariance matrix are computed directly by averaging the products of pairs of coefficients over all the neighborhoods in the subband.

6.2.3 Estimation of Image in Noise

The task of the least square optimal Bayesian estimator is to estimate the original transform coefficients from a noisy observation. The observation vector \mathbf{y} corresponding to a neighborhood of N observed transform coefficients can be expressed as

$$\mathbf{y} = \mathbf{x} + \mathbf{w} = \sqrt{z}\mathbf{u} + \mathbf{w}, \quad (6.7)$$

where \mathbf{x} and \mathbf{w} are neighborhoods of the signal and noise transform coefficients, respectively and \mathbf{x} is given by (6.1). Notice that the three random variables on the right-hand side of (6.7) are independent and both \mathbf{u} and \mathbf{w} are zero-mean Gaussian random vectors, with associated covariance matrices \mathbf{C}_u and \mathbf{C}_w . The density of the observed neighborhood vector conditioned on z is zero-mean Gaussian, with the covariance $\mathbf{C}_{\mathbf{y}|z} = z\mathbf{C}_u + \mathbf{C}_w$, and can be expressed as

$$p(\mathbf{y}|z) = \frac{\exp\left(\frac{-\mathbf{y}^T(z\mathbf{C}_u + \mathbf{C}_w)^{-1}\mathbf{y}}{2}\right)}{\sqrt{(2\pi)^N |z\mathbf{C}_u + \mathbf{C}_w|}}. \quad (6.8)$$

Given \mathbf{C}_w , the covariance \mathbf{C}_u can be computed from the observation covariance \mathbf{C}_y . We compute \mathbf{C}_y from $\mathbf{C}_{\mathbf{y}|z}$ by taking expectations over z , that is,

$$\mathbf{C}_y = E\{z\}\mathbf{C}_u + \mathbf{C}_w.$$

Setting $E\{z\} = 1$, we obtain

$$\mathbf{C}_u = \mathbf{C}_y - \mathbf{C}_w. \quad (6.9)$$

For the reasons of numerical imprecisions, some eigenvalues of \mathbf{C}_u might be negative. We set them to zero forcing \mathbf{C}_u to be positive semidefinite.

For each neighborhood, we estimate the reference transform coefficient x_c at the center of the neighborhood using the Bayes least square estimation from the neighborhood vector \mathbf{y} . The Bayes least square estimation is the conditional mean, that is, it is given by

$$\begin{aligned} E\{x_c|\mathbf{y}\} &= \int x_c p(x_c|\mathbf{y}) dx_c = \int \int_0^\infty x_c p(x_c, z|\mathbf{y}) dz dx_c \\ &= \int \int_0^\infty x_c p(x_c|\mathbf{y}, z) p(z|\mathbf{y}) dz dx_c = \int_0^\infty E\{x_c|\mathbf{y}, z\} p(z|\mathbf{y}) dz, \end{aligned} \quad (6.10)$$

where we assume uniform convergence in order to exchange the order of integration. Thus, the solution is the average of the Bayes least squares estimate of x_c when conditioned on z , weighted by the posterior density $p(z|\mathbf{y})$. We now analyze each of the two components in the integral in (6.10).

The first component $E\{x_c|\mathbf{y}, z\}$ is the *local Wiener estimate* as the neighborhood vector \mathbf{x} is Gaussian when conditioned on z . The estimate for the full neighborhood vector \mathbf{x} is given by

$$E\{\mathbf{x}|\mathbf{y}, z\} = z\mathbf{C}_u (z\mathbf{C}_u + \mathbf{C}_w)^{-1} \mathbf{y}. \quad (6.11)$$

We diagonalize the matrix $z\mathbf{C}_u + \mathbf{C}_w$ in order to simplify the dependence of $E\{\mathbf{x}|\mathbf{y}, z\}$ on the multiplier z . Let \mathbf{S} be the symmetric square root of the positive definite matrix \mathbf{C}_w , that is, $\mathbf{C}_w = \mathbf{S}\mathbf{S}^T$. Furthermore, let $\{\mathbf{Q}, \Lambda\}$ be the eigenvector/eigenvalue expansion of the matrix $\mathbf{S}^{-1}\mathbf{C}_u\mathbf{S}^{-T}$. Then

$$\begin{aligned} z\mathbf{C}_u + \mathbf{C}_w &= z\mathbf{C}_u + \mathbf{S}\mathbf{S}^T = \mathbf{S} (z\mathbf{S}^{-1}\mathbf{C}_u\mathbf{S}^{-T} + \mathbf{I}) \mathbf{S}^T \\ &= \mathbf{S}\mathbf{Q} (z\Lambda + \mathbf{I}) \mathbf{Q}^T \mathbf{S}^T. \end{aligned} \quad (6.12)$$

Now the expectation in (6.11) becomes

$$\begin{aligned} E\{\mathbf{x}|\mathbf{y}, z\} &= z\mathbf{C}_u\mathbf{S}^{-T}\mathbf{Q}(z\mathbf{\Lambda} + \mathbf{I})^{-1}\mathbf{Q}^T\mathbf{S}^{-1}\mathbf{y} \\ &= z\mathbf{S}\mathbf{S}^{-1}\mathbf{C}_u\mathbf{S}^{-T}\mathbf{Q}(z\mathbf{\Lambda} + \mathbf{I})^{-1}\mathbf{Q}^T\mathbf{S}^{-1}\mathbf{y} \\ &= z\mathbf{S}\mathbf{Q}\mathbf{\Lambda}(z\mathbf{\Lambda} + \mathbf{I})^{-1}\mathbf{Q}^T\mathbf{S}^{-1}\mathbf{y} = z\mathbf{M}\mathbf{\Lambda}(z\mathbf{\Lambda} + \mathbf{I})^{-1}\mathbf{v}, \end{aligned} \quad (6.13)$$

where $\mathbf{M} = \mathbf{S}\mathbf{Q}$, and $\mathbf{v} = \mathbf{M}^{-1}\mathbf{y}$. Finally, we restrict the estimate to the reference coefficient x_c , as needed for the solution of (6.10), that is, we compute

$$E\{x_c|\mathbf{y}, z\} = \sum_{n=1}^N \frac{zm_{cn}\lambda_n v_n}{z\lambda_n + 1}, \quad (6.14)$$

where $m_{i,j}$ represents an element of the matrix \mathbf{M} in the i th row and j th column, λ_n are the diagonal elements of $\mathbf{\Lambda}$, v_n are the elements of the vector \mathbf{v} , and c is the index of the reference coefficient within the neighborhood vector.

The second component in the integral in (6.10) is distribution of the multiplier z conditioned on the observed neighborhood vector \mathbf{y} . According to the Bayes' rule, we have

$$p(z|\mathbf{y}) = \frac{p(\mathbf{y}|z)p_z(z)}{\int_0^\infty p(\mathbf{y}|\alpha)p_z(\alpha)d\alpha}. \quad (6.15)$$

The conditional probability density $p(\mathbf{y}|z)$ in (6.15) is given by (6.8). After the simplification using (6.12) and using the Jeffrey's prior given by (6.5), $p(\mathbf{y}|z)$ becomes

$$p(\mathbf{y}|z) = \frac{\exp\left(-\frac{1}{2}\sum_{n=1}^N \frac{v_n^2}{z\lambda_n+1}\right)}{\sqrt{(2\pi)^N |\mathbf{C}_w| \prod_{n=1}^N (z\lambda_n + 1)}}. \quad (6.16)$$

Finally, we are able to compute numerically the integral in the Bayes least square estimation given by (6.10).

6.2.4 Image Denoising Algorithm

The image denoising algorithm that uses the GSM model, as proposed in [68], consists of three main parts: (1) transform of the image, (2) denoising of each subband, except the low-pass (LP) residual, and (3) inverse transform of the denoised coefficients. The steerable pyramids explained in [85] are used as the transform in this method.

The complete GSM-based denoising algorithm proposed in [68] is summarized in Algorithm 6.1.

6.3 Directionlets-based Image Denoising

Oversampled transforms have been shown to lead to better denoising performance than critically sampled transforms [20]. For that reason, the denoising algorithm we propose in this chapter is based on the undecimated directionlets. Oversampling is imposed in the same way as in the case of the undecimated wavelet transform, that is, by discarding the subsamplers

Algorithm 6.1 GSM-based denoising algorithm proposed in [68].

Step 1: Forward transform

Decompose the image into subbands.

Step 2: Estimation

For each subband, except the LP residual:

- Compute the neighborhood noise covariance \mathbf{C}_w from the standard deviation in the original domain calculated using (6.6);
- Estimate noisy neighborhood covariance \mathbf{C}_y ;
- Estimate \mathbf{C}_u from \mathbf{C}_w and \mathbf{C}_y using (6.9);
- Compute Λ and \mathbf{M} ;
- For each neighborhood:
 - For each value z in the integration range:
 - Compute $E\{x_c|\mathbf{y}, z\}$ using (6.14);
 - Compute $p(\mathbf{y}|z)$ using (6.16);
 - Compute $p(z|\mathbf{y})$ using (6.15) and (6.5);
 - Compute $E\{x_c|\mathbf{y}\}$ numerically using (6.10);

Step 3: Inverse transform

Reconstruct the denoised image from the processed subbands and the LP residual.

in the filter-banks [32; 101]. Therefore, all the transform subbands have the same number of coefficients as in the original domain.

The GSM estimation outlined in Algorithm 6.1 is applied on these transform coefficients. However, the shape of the neighborhoods is changed in our method. We distinguish between two types of denoising in the method:

- (1) The denoising performed by estimation from an *isotropic neighborhood* of the coefficients obtained using the *isotropic WT* (which we call *smooth denoising*),
- (2) The denoising performed by estimation from an *anisotropic neighborhood* of the coefficients obtained using the *anisotropic WT*, where both the transform and neighborhood have the same orientation (which we call *oriented denoising*).

Intuitively, oriented denoising achieves a good estimation in the regions of images with a strong directionality matched with the orientation of the transform and neighborhood. The oriented neighborhoods capture more efficiently the oriented features in images that consist of pixels with similar statistical properties aligned along the dominant direction. Similarly, smooth denoising performs better in smooth regions without a clear dominant direction.

In the sequel, we first explain the construction of both the isotropic and anisotropic neighborhoods. Then, we present some details on the smooth and oriented denoising methods with denoised versions of a toy example image. Finally, we show a denoising method based on the linear combination of the smooth and oriented denoised versions in the original pixel domain and experiments with some natural images.

6.3.1 Isotropic and Anisotropic Neighborhoods

Isotropy and orientation of neighborhoods are chosen in such a way that the transform coefficients with similar statistical characteristics are grouped together and centered around the referent coefficient. Furthermore, all neighborhoods, except the ones at the coarsest scale, include the parent coefficient, that is, the coefficient at the next coarser scale in the same spatial location as the referent coefficient. Experiments in [68] and in Section 6.4 show that this inclusion contributes to better denoising performance.

Recall that the filters used in the undecimated WT are upsampled across scales. Thus, if the LP filter $H_0(z)$ is used in the corresponding 1-D wavelet filter-bank, then the equivalent LP filter at the j th scale is $H_0(z^{J-j})$, where $j = J, \dots, 1$ corresponds to the scale index sweeping from the finest to the coarsest scale. Notice that, since the subsamplers are discarded in the undecimated WT, the transform coefficients in the neighbor spatial locations are highly correlated. Therefore, estimation based on the closest neighborhood is not too efficient. For that reason, we choose sparser neighborhoods, that is, the neighborhoods of coefficients with the relative coordinates to the coordinate of the referent coefficient scaled by the distance factor

$$\delta = 2^{J-j}, \quad (6.17)$$

while the dimensionality of the neighborhood is retained the same.

The isotropic neighborhood is used along with the isotropic WT in smooth denoising. The neighborhood around the referent coefficient in the spatial location (c_1, c_2) consists of the parent and the coefficients at the same scale in the spatial locations (c'_1, c'_2) (see Figure 6.2(a)), where

$$c'_1 = c_1 + n\delta, c'_2 = c_2 + m\delta, n, m \in \{-1, 0, 1\}, \quad (6.18)$$

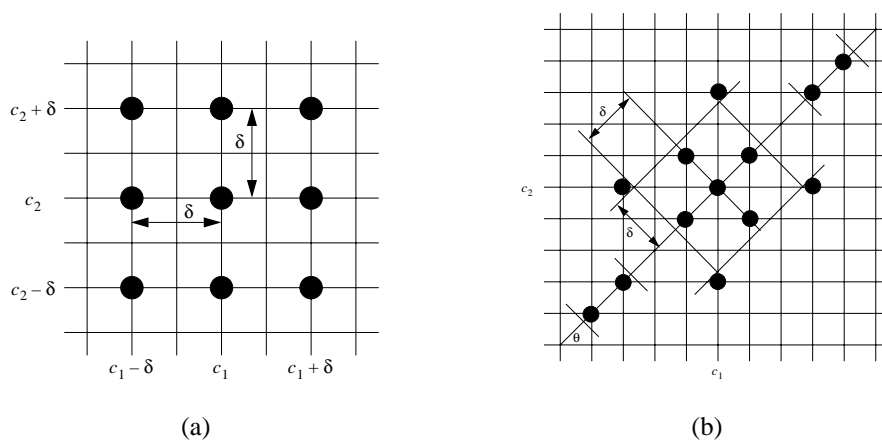


Figure 6.2: Two types of neighborhood used in the estimation method. (a) The isotropic neighborhood used along with the isotropic WT in smooth denoising. The neighborhood is centered around the referent coefficient in the spatial location (c_1, c_2) and includes the coefficients in the spatial locations $(c_1 \pm \delta, c_2 \pm \delta)$, where δ is the distance factor given by (6.17), and the corresponding parent. (b) The anisotropic neighborhood is used along with the anisotropic WT in oriented denoising. The orientation of the neighborhood is matched with the orientation of the transform. The included coefficients are located in the closest integer coordinates to the coordinates in the rotated system for 7 centered coefficients along the dominant direction and 3 coefficients in the next parallel lines. The shown example is constructed for the values $\theta = -45^\circ$ and $\delta = 2$.

Here, δ is given by (6.17).

The anisotropic neighborhood used in oriented denoising have the orientation matched with the orientation of the WT. Given the orientation θ of the transform, the neighborhood around the coefficient in the location (c_1, c_2) is defined as the union of the corresponding parent and the set of the coefficients at the same scale in the locations (c'_1, c'_2) , where

$$\begin{aligned} c'_1 &= [(c_1 + n\delta) \cos \theta - (c_2 + m\delta) \sin \theta] \\ c'_2 &= [(c_1 + n\delta) \sin \theta + (c_2 + m\delta) \cos \theta], \end{aligned} \quad (6.19)$$

for $n \in \{-3, \dots, 3\}$ if $m = 0$, and $n \in \{-1, 0, 1\}$ if $m \in \{-1, 1\}$. Here, $[x]$ stands for the closest rounded integer to x .

Notice that the rounding functions used in (6.19) ensure that the coordinates of the neighbor coefficients are integers. In general, there are 13 coefficients at the same scale and 1 parent coefficient in an anisotropic neighborhood. However, for some values of θ , two or more coefficients defined by (6.19) might overlap generating a smaller neighborhood. The anisotropic neighborhood for the case of $\delta = 2$ and $\theta = -45^\circ$ is illustrated in Figure 6.2(b).

6.3.2 Smooth Denoising

Smooth denoising is performed by estimation using the GSM model in the transform domain, where the undecimated isotropic transforms are applied. In the experiments in Section 6.4 the S-AWT(I,1,1) and S-AWT(Q,1,1) are used, where **I** is the identity matrix and

$$\mathbf{Q} = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}.$$

Notice that these two transforms correspond to the standard undecimated 2-D WT along the standard ($0^\circ/90^\circ$) and diagonal ($45^\circ/-45^\circ$) directions.

The isotropic neighborhood of transform coefficients is used in estimation in the case of both transforms. The smooth denoised output is obtained as the mean of the two reconstructed images.

Since the estimation of referent coefficients is based on the statistical characteristics of the coefficients in a neighborhood without a dominant orientation, smooth denoising achieves better performance in the case of smooth regions in images. On the other hand, the coefficients around edges are estimated from the neighborhoods that consist of coefficients from both sides of the edges with significantly different statistics. Therefore, the quality of estimation is reduced and visual artifacts appear around edges.

An example is shown in Figure 6.3, where a piecewise smooth image is denoised by smooth denoising. The numerical comparisons of the images to the original version are given in terms of *peak-signal-to-noise-ratio* (PSNR) factor equal to $20 \log_{10}(255/\sigma_e)$, where σ_e is the standard deviation of the error. The reconstructed image is efficiently denoised in the smooth regions, but the artifacts around the edges are noticeable, as shown in Figure 6.3(c).

6.3.3 Oriented Denoising

In oriented denoising, the orientation of neighborhoods is matched with the orientation of the transform. In this way, edges are better preserved in the denoised images. Some other denoising methods also perform the edge-preservation processing [54; 115].

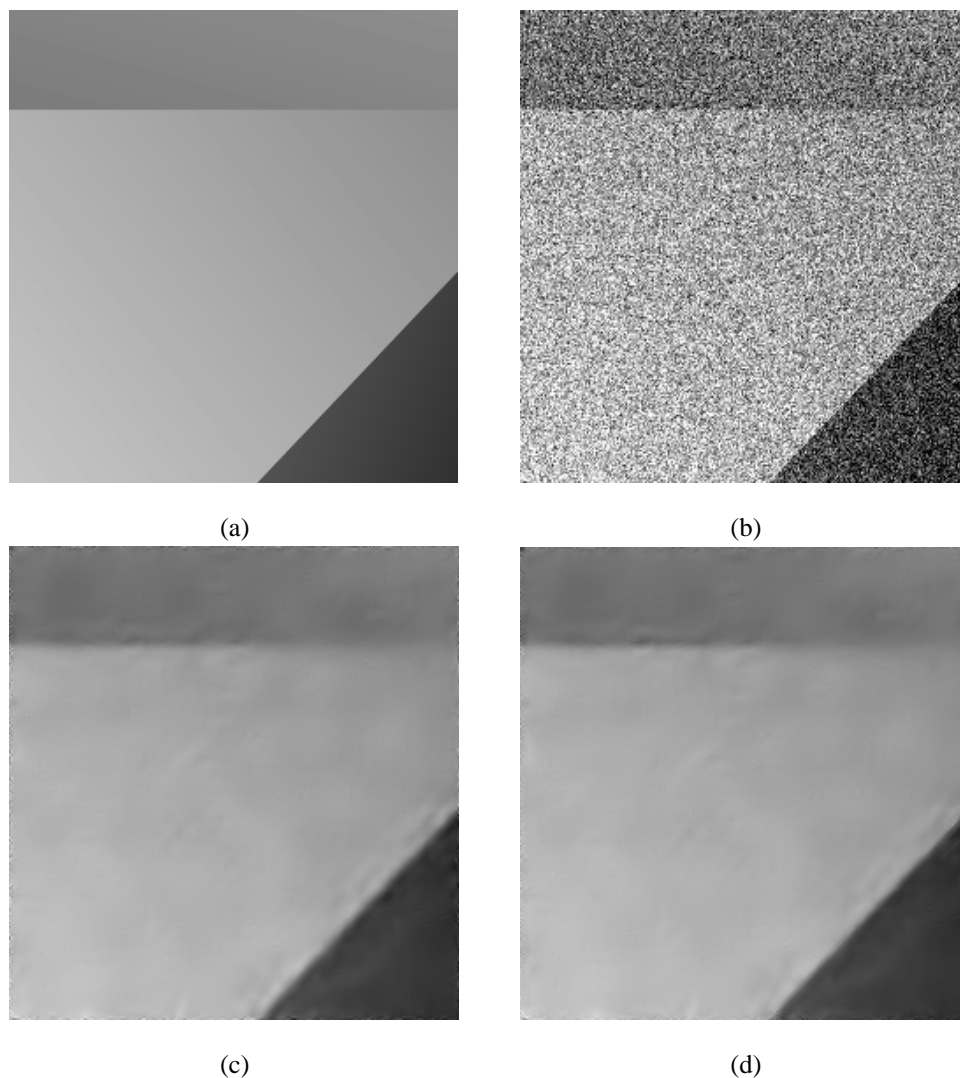


Figure 6.3: A synthetic image and the efficiency of smooth denoising. (a) Original piecewise smooth image with linear discontinuities. (b) A noisy version of the image with $\sigma = 50$, i.e. PSNR= 14.12dB. (c) Smooth denoised output obtained by estimation using the isotropic neighborhoods and the isotropic undecimated transforms S-AWT(I,1,1) and S-AWT(Q,1,1) that correspond to the standard undecimated 2-D WT along the standard and diagonal directions. The smooth regions are denoised efficiently, but the regions around edges contain significant visual artifacts. The numerical quality of the denoised image is PSNR=32.93dB. (d) The denoised version obtained by the linear combination of the smooth and oriented denoised outputs. The numerical quality is PSNR=33.57dB. The visual quality of the edges is improved.

In the experiments a set of four anisotropic transforms S-AWT($M_i, 2, 1$) is used, for $i = 1, 2, 3, 4$. The corresponding generator matrices are defined as

$$\mathbf{M}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \mathbf{M}_2 = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}, \mathbf{M}_3 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \mathbf{M}_4 = \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}. \quad (6.20)$$

Notice that the anisotropic transforms used in the experiments are oriented along 0° , 45° , 90° , and -45° , respectively.

The oriented denoised outputs achieve better results around edges with orientation close to the orientation of the transform. The reason for this good performance is in an efficient capturing of the coefficients with similar statistical characteristics in a neighborhood used in estimation. However, the denoising results in smooth regions are not as good as in the case of smooth denoising.

The oriented denoised outputs of the piecewise smooth image shown in Figure 6.3(a) are presented in Figure 6.4. Notice that even though the numerical results of the denoised images are weaker when compared to that in the case of smooth denoising, the visual artifacts around edges are suppressed.

Notice also that we use only 4 directions here. Recall that the construction of directionlets allows for more transform directions, but with increasing computational complexity. For that reason, we retain only 4 transform directions as a good trade-off between complexity and denoising efficiency.

6.3.4 Combined Smooth and Oriented Denoising

As shown in Sections 6.3.2 and 6.3.3, smooth denoising achieves good results in smooth regions without edges, whereas oriented denoising dominates in the regions around edges. Furthermore, oriented denoising is better in the case of matching (exact or approximate) between the dominant direction in the image and the transform direction. In this section, we combine these two types of denoised outputs in the pixel domain to exploit the advantages of both.

The key point to improve the denoising results by the combination of the two denoising methods is an efficient detection of local directionality in the image. For each pixel, we determine if it belongs to a smooth or edge region, and, in the latter case, the dominant direction from a predetermined set.

Recall that 4 oriented denoised outputs along the directions 0° , 90° , 45° , and -45° are generated in the experiments in Section 6.3.3. We call them y_{d_1} , y_{d_2} , y_{d_3} , and y_{d_4} , respectively. In order to determine the directionality of each pixel, we filter each oriented output y_{d_i} along the matched direction d_i by the upsampled version $H_1(z^4)$ of the high-pass (HP) filter $H_1(z)$ from the "9-7" filter-bank. The 4 produced subbands we denote as \hat{y}_{d_i} , for $i = 1, \dots, 4$. This upsampling factor for the HP filter is hand-optimized as a good trade-off between locality of the directionality information and strong correlation among neighbor pixels in images. The results of the HP filtering along the 4 directions for the toy image shown in Figure 6.3(a) is presented in Figure 6.5.

The dominant direction $d_{dom}(n, m)$ for a pixel in the location (n, m) is the transform direction that minimizes the magnitudes of the coefficients $\hat{y}_{d_i}(n, m)$, that is,

$$d_{dom}(n, m) = \arg \min_{d_i} |\hat{y}_{d_i}(n, m)|. \quad (6.21)$$

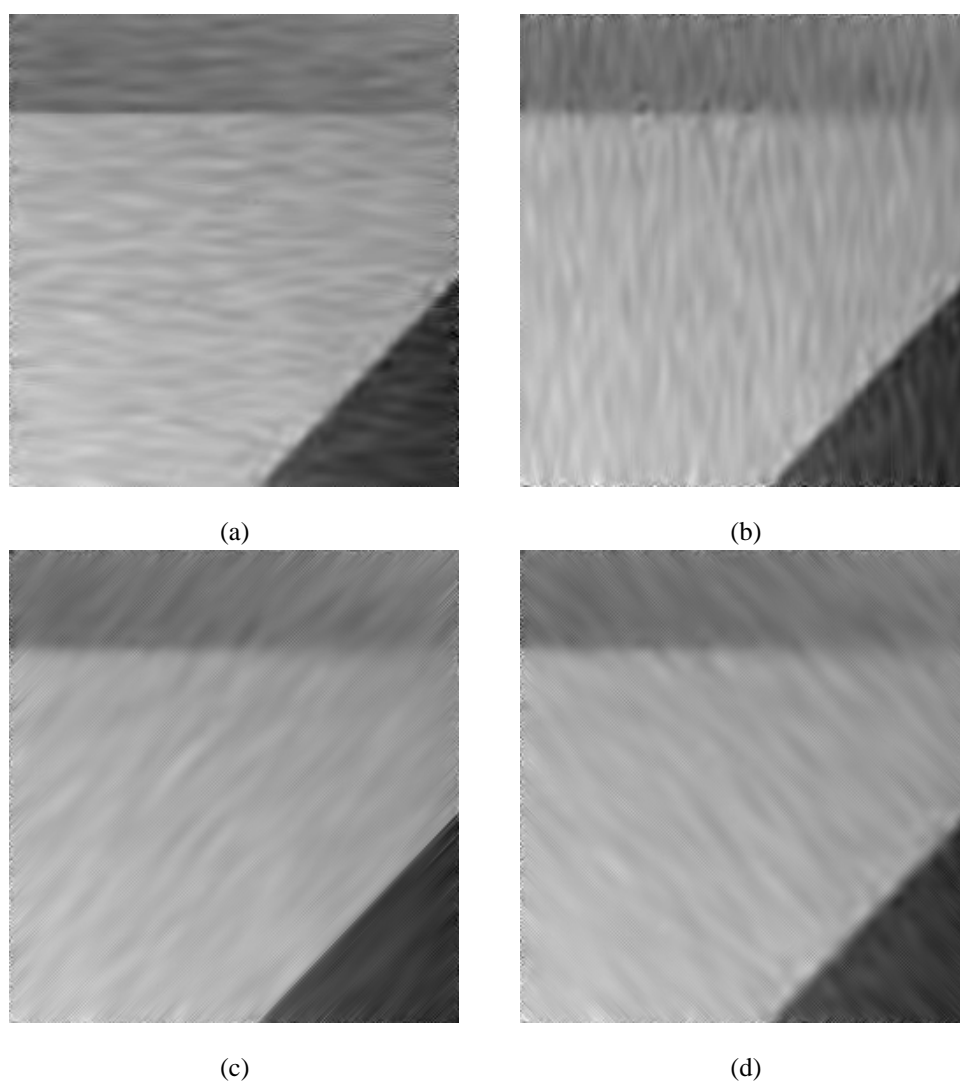


Figure 6.4: Oriented denoised outputs of the noisy image shown in Figure 6.3(b) obtained using anisotropic transforms and anisotropic neighborhoods with matched directions along (a) 0° , (b) 90° , (c) -45° , and (d) 45° . The PSNR factors are: (a) 30.74dB, (b) 30.16dB, (c) 30.47dB, and (d) 29.56dB. Even though the numerical results are weaker than the results obtained in smooth denoising, the visual artifacts around the edges with matched directions are reduced. This happens because the coefficients with similar statistical characteristics are grouped in the estimation phase.

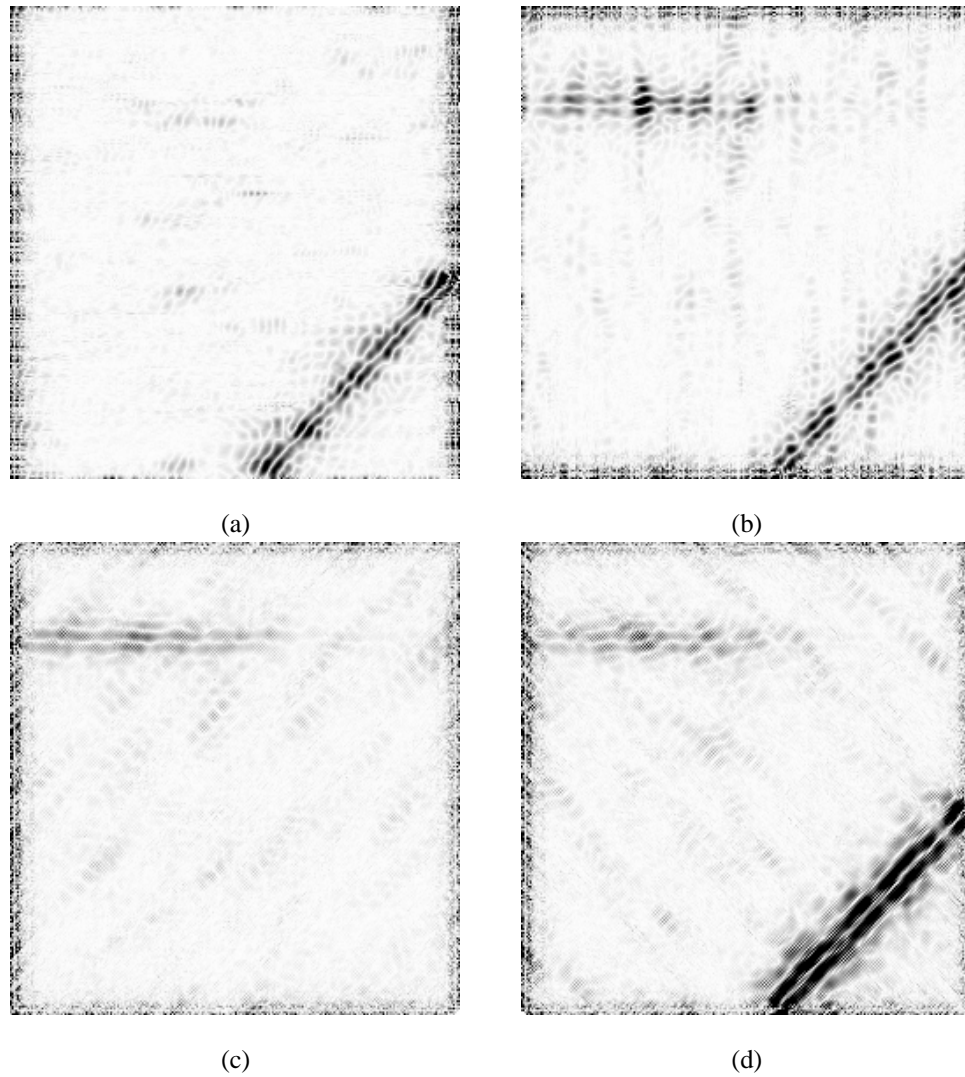


Figure 6.5: The oriented denoised outputs shown in Figure 6.4 are filtered using the HP filter $H_1(z^4)$ along 4 directions. The magnitudes of the coefficients are shown for (a) 0° , (b) 90° , (c) -45° , and (d) 45° . For each spatial location, the minimal magnitude indicates the dominant direction.

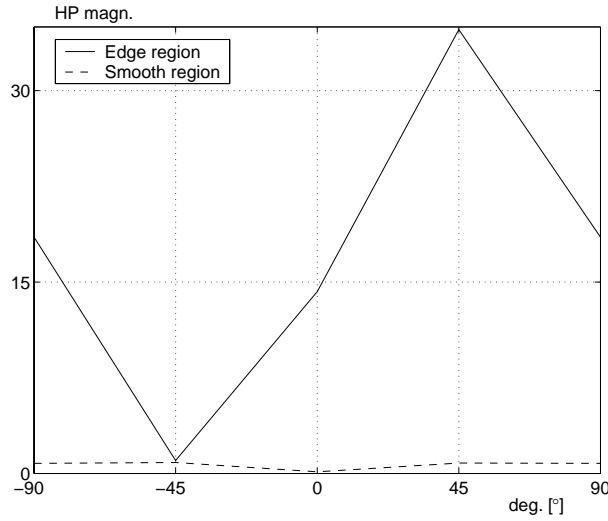


Figure 6.6: The oriented denoised outputs are filtered using the upsampled HP filters to determine if each pixel belongs to a smooth or edge region and its dominant direction. The typical fluctuations of the energy of the HP coefficients across directions are shown for the case of a smooth and edge pixel. The fluctuation in the case of smooth pixels is significantly smaller than in the case of edge pixels. The small difference $\Delta(n, m)$ defined by (6.22) for the function shown by the dotted line indicates that the pixel belongs to a smooth region.

Since the coefficients in \hat{y}_{d_i} are produced by HP filtering, the local neighborhood around the location (n, m) is the smoothest along the direction $d_{dom}(n, m)$ and, thus, this direction approximates best the local dominant direction in the location (n, m) .

Notice that $d_{dom}(n, m)$ can be also determined for the pixels in smooth regions because of small fluctuations of the denoised outputs across directions (see Figure 6.6 for the comparison between these functions in a smooth and edge region). Since these pixels are better denoised by the smooth denoised output, the detection of dominant direction in this case is not valuable. In order to distinguish between the smooth and edge regions, we measure these fluctuations by computing the difference between the mean and minimum of the magnitudes across directions for each pixel, that is, we compute

$$\Delta(n, m) = \frac{1}{4} \sum_{i=1}^4 |\hat{y}_{d_i}(n, m)| - |\hat{y}_{d_{dom}(n, m)}(n, m)|. \quad (6.22)$$

A small difference $\Delta(n, m)$ indicates that the location (n, m) belongs to a smooth region, whereas a large value implies an edge region. Since the value of $\Delta(n, m)$ is influenced by very complex interdependencies among pixels in images, there does not exist an exact threshold which determines if a location belongs to a smooth or edge region. For that reason, we use a soft transition between smooth and edge regions introducing a linear combination of the smooth denoised pixels y_s and the best (in the sense given by (6.21)) oriented denoised

pixels $y_{d_{dom}(n,m)}$. The final denoised pixel in the location (n, m) is defined by

$$y(n, m) = \alpha(n, m)y_s(n, m) + (1 - \alpha(n, m))y_{d_{dom}(n,m)}(n, m), \quad (6.23)$$

where $0 \leq \alpha(n, m) \leq 1$. The coefficient $\alpha(n, m)$ is defined by

$$\alpha(n, m) = \begin{cases} 1 & \Delta(n, m) \leq 1, \\ -\frac{1}{14}\Delta(n, m) + \frac{15}{14} & 1 \leq \Delta(n, m) \leq 15, \\ 0 & \Delta(n, m) \geq 15 \end{cases} \quad (6.24)$$

The threshold values 1 and 15 used in (6.24) are hand-optimized. Notice that the thresholds, like 3σ or the universal threshold [32], that are obtained by minimization of the impact of the noise are not appropriate in our case because the pixels are already denoised and the distribution of the retained noise is not Gaussian anymore. That makes very difficult to have an efficient optimization of the thresholds in (6.24). Fortunately, the result of denoising is not too sensitive to the choice of these thresholds.

The choice of denoised pixels is shown in Figure 6.7 for the piecewise smooth image from Figure 6.3(a). The 4 gray-level images in Figure 6.7(a)-(d) related to the oriented denoised outputs represent the magnitude of the inverse coefficient $1 - \alpha$, whereas the image in Figure 6.7(e) related to the smooth denoised output illustrates the magnitude of the coefficient α .

The denoised image produced by the linear combination is shown in Figure 6.3(d). The PSNR factor is improved and is equal to 33.57dB. The visual quality of the edges in the denoised version is also improved comparing to the smooth denoised version of the toy example.

The whole denoising algorithm is outlined in Algorithm 6.2.

6.4 Numerical Experiments

We perform the denoising experiments on the same natural images that have been used in [68] and we compare the obtained results to those in [68].

The results of denoising of the test image Barbara are presented in Figures 6.8 and 6.9. The image Barbara is corrupted by the additive white Gaussian noise with different input noise powers σ^2 (or the corresponding PSNR factors). The denoising method based on directionlets is compared to the method that uses the standard undecimated 2-D WT and the steerable pyramids [68]. The results are shown in Figure 6.8. The original and noisy version of the image are shown in Figure 6.9(a) and (b) for PSNR=8.13dB. The two reconstructed versions made by the standard undecimated 2-D WT and directionlets are presented in Figure 6.9(c) and (d) for PSNR=21.66dB and PSNR=22.06dB, respectively. Notice that the denoised image using directionlets has sharper edges than the one denoised by the standard transform because the pixels in the regions around edges are obtained by oriented denoising.

The numerical results of the denoising methods based on directionlets and steerable pyramids applied on all the test images used in [68] are outlined in Table 6.1. Notice that the numerical results obtained by the steerable pyramids are slightly better than the results based on directionlets. However, the transform in the case of directionlets is computationally simpler than the steerable pyramids. Recall that, in directionlets, only the separable filtering using the 1-D filter-banks is taking place, unlike in the case of steerable pyramids, where 2-D directional filter-banks are used [85].

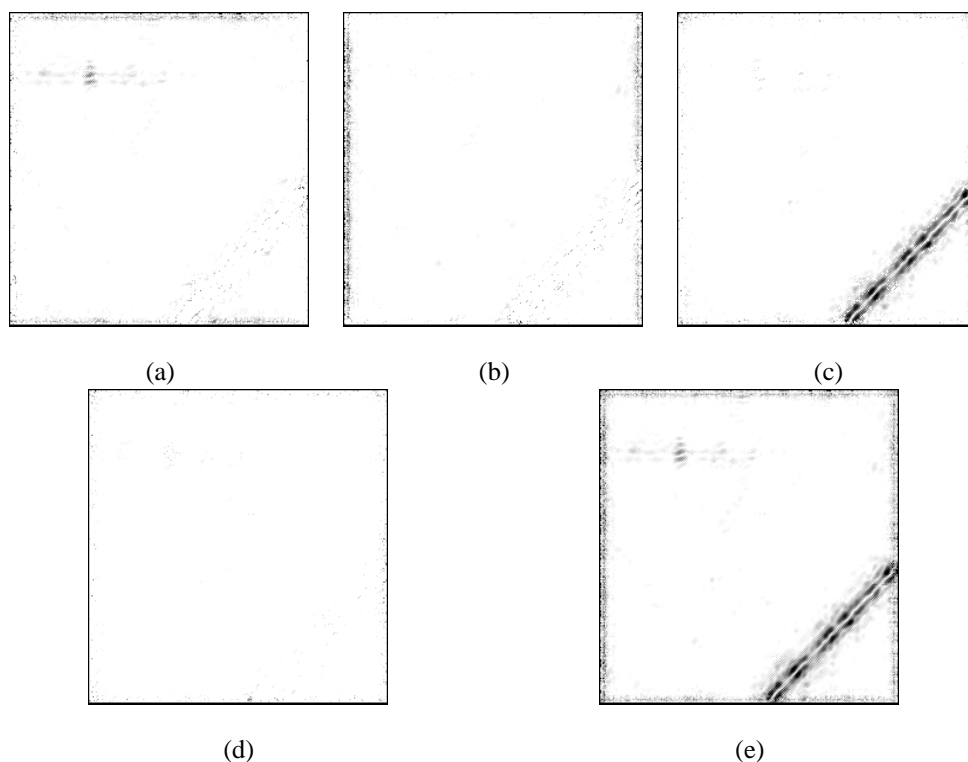


Figure 6.7: The best oriented denoised pixels are chosen from the directional sub-band $\hat{y}_{d_{\text{dom}}}$ that minimizes the magnitude of the HP coefficient for each spatial location (n, m) . These pixels are linearly combined with the pixels in the smooth denoised output y_s using the coefficient α . The coefficient α is computed using a soft function of the difference between the mean and minimum of the HP energy across directions in the oriented denoised outputs, as given by (6.24). The values of α are in the interval $[0, 1]$. The inverse values $1 - \alpha$ are shown in gray-scale images for the oriented denoised outputs along (a) 0° , (b) 90° , (c) -45° , (d) 45° . (e) The magnitudes of α for the smooth denoised output.

Algorithm 6.2 Combined denoising algorithm using smooth and oriented denoising outputs.

Step 1: Smooth denoising

- Compute S-AWT($\mathbf{I}, 1, 1$) and S-AWT($\mathbf{Q}, 1, 1$);
- Apply GSM estimation outlined in Algorithm 6.1 using the isotropic neighborhoods;
- Compute the inverse S-AWT($\mathbf{I}, 1, 1$) and S-AWT($\mathbf{Q}, 1, 1$);
- Compute y_s as the mean of the two reconstructions.

Step 2: Oriented denoising

- Compute S-AWT($\mathbf{M}_i, 2, 1$), for $i = 1, \dots, 4$, where \mathbf{M}_i are given by (6.20);
- Apply GSM estimation outlined in Algorithm 6.1 using the anisotropic neighborhoods;
- Compute y_{d_i} as the inverse S-AWT($\mathbf{M}_i, 2, 1$), for $i = 1, \dots, 4$.

Step 3: Combined denoising

For each spatial location (n, m) :

- Filter y_{d_i} in the direction d_i using the filter $H_1(z^4)$;
- Compute $d_{dom}(n, m)$ using (6.21);
- Compute $\Delta(n, m)$ using (6.22);
- Compute $\alpha(n, m)$ using (6.24);
- Compute the final denoised pixel $y(n, m)$ using (6.23).

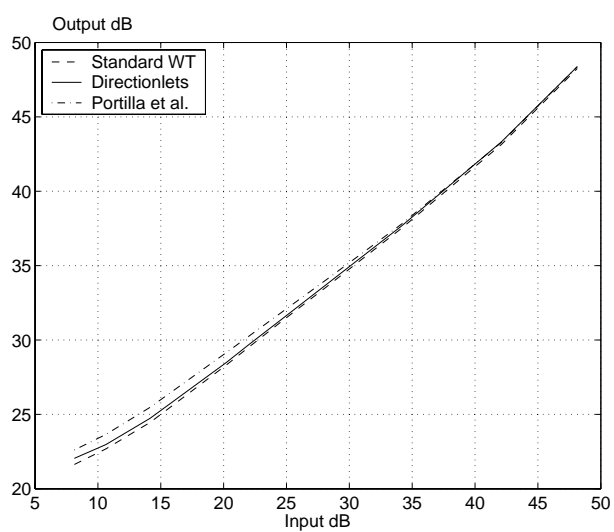


Figure 6.8: The test image Barbara is corrupted by the additive white Gaussian noise with different power of the noise. The image is denoised using the estimation in the transform domain of three transforms: the standard undecimated 2-D WT, the steerable pyramids, and directionlets. The method based on directionlets outperforms the method that uses the standard undecimated 2-D WT, but it is weaker than the steerable pyramids. However, the separable filtering used in the case of directionlets yields a lower computational complexity than in the case of non-separable 2-D filtering used in the steerable pyramids.

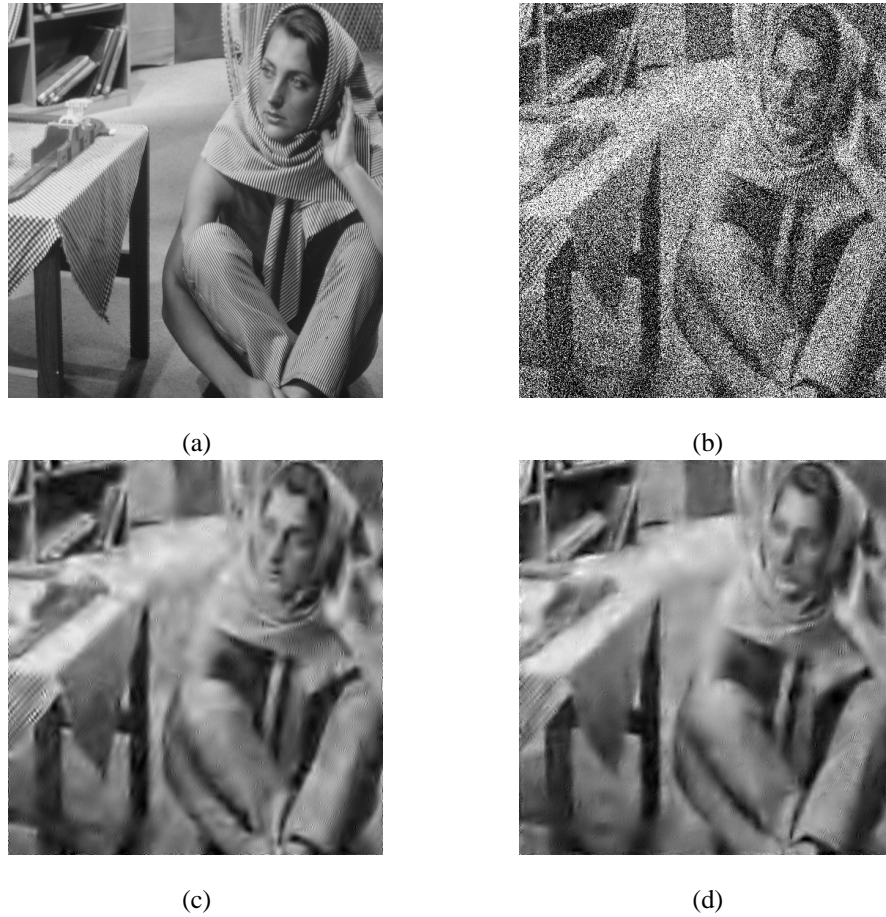


Figure 6.9: The test image Barbara is corrupted by the additive white Gaussian noise. The noisy image is denoised by estimation in the transform domain, where the two transforms are applied: the standard undecimated 2-D WT and directionlets. Both the numerical and visual quality of the denoised version obtained by directionlets is better than the version obtained by the standard transform. (a) Original image. (b) Noisy version at PSNR=8.13dB. (c) Denoised image obtained by the standard undecimated WT at PSNR=21.66dB. (d) Denoised image obtained by directionlets at PSNR=22.06dB.

Table 6.1: Denoising results expressed in terms of PSNR. The results are shown for two methods based on directionlets and steerable pyramids. The latter method is explained in [68].

Directionlets						
σ / PSNR	<i>Lena</i>	<i>Barbara</i>	<i>Boats</i>	<i>Fngprnt</i>	<i>House</i>	<i>Peppers</i>
1 / 48.13	48.46	48.38	48.40	48.38	48.91	48.46
2 / 42.11	43.26	43.33	43.00	42.84	44.16	43.14
5 / 34.15	38.40	37.66	36.98	36.17	38.59	37.51
10 / 28.13	35.22	33.70	33.38	31.61	35.07	33.59
15 / 24.61	33.31	31.39	31.31	29.22	33.27	31.35
20 / 22.11	31.91	29.76	29.86	27.62	31.86	29.72
25 / 20.17	30.80	28.46	28.76	26.46	30.77	28.40
50 / 14.15	27.59	24.74	25.71	23.15	27.37	25.13
75 / 10.63	26.01	22.97	24.10	21.36	25.76	23.56
100 / 8.13	24.94	22.05	23.17	20.14	24.27	22.22

Steerable pyramids						
σ / PSNR	<i>Lena</i>	<i>Barbara</i>	<i>Boats</i>	<i>Fngprnt</i>	<i>House</i>	<i>Peppers</i>
1 / 48.13	48.46	48.37	48.44	48.46	48.85	48.38
2 / 42.11	43.23	43.29	42.99	43.05	44.07	43.00
5 / 34.15	38.49	37.79	36.97	36.68	38.65	37.31
10 / 28.13	35.61	34.03	33.58	32.45	35.35	33.77
15 / 24.61	33.90	31.86	31.70	30.14	33.64	31.74
20 / 22.11	32.66	30.32	30.38	28.60	32.39	30.31
25 / 20.17	31.69	29.13	29.37	27.45	31.40	29.21
50 / 14.15	28.61	25.48	26.38	24.16	28.26	25.90
75 / 10.63	26.84	23.65	24.79	22.40	26.41	24.00
100 / 8.13	25.64	22.61	23.75	21.22	25.11	22.66

The denoising results obtained using directionlets are also competitive to the other *state-of-the-art* denoising results, like the ones obtained by curvelets [87] and contourlets [23]. However, similarly to the case of the steerable pyramids, these transforms are computationally more complex than directionlets because of non-separable 2-D filtering.

6.5 Conclusions

In this chapter, we have shown that the undecimated directionlets are very efficient tool for image denoising. The GSM-based estimation model in the transform domain achieves the denoising results competitive to the state-of-the-art results, while the complexity of the transform is lower because of the use of separable filter-banks.

Chapter 7

Conclusion

7.1 Summary

In this thesis, we have analyzed the problem of efficient representation of geometrical features in images that are highly anisotropic and have certain orientations, in general, different from horizontal and vertical. The representation of these features provided by the standard 2-D wavelet transform is non-sparse since they intersect with many wavelets producing high magnitude transform coefficients. We have proposed an integer lattice-based transform construction that leads to anisotropic basis functions with directional vanishing moments in arbitrary directions. We call these critically sampled basis functions directionlets. We have shown that directionlets achieve better non-linear approximation and compression performance than the standard method and are comparable to the other oversampled transforms. We have also presented an efficient image denoising method based on the undecimated version of directionlets. The main contributions of the thesis are summarized as follows.

Skewed anisotropic transform constructions

In Chapters 2 and 3, we have studied the limits of the standard 2-D wavelet transform in providing an efficient representation of images with elongated objects, like edges and contours, in different directions, not only horizontal and vertical. We have proposed two new lattice-based anisotropic transforms, which are called skewed fully separable and skewed anisotropic wavelet transforms. These transforms lead to basis functions, which we call directionlets. The transforms retain the simplicity of separable filtering and subsampling from the standard 2-D wavelet transform, while imposing directional vanishing moments in directionlets along two arbitrary directions with rational slopes. The directional vanishing moments allow for an efficient capturing of the geometrical features in images.

General multi-directional transforms

In Chapter 4, we have presented the construction of a general multi-directional transforms. These transforms retain separability of the standard 2-D wavelet transform and allow the corresponding basis functions to have directional vanishing moments along arbitrary set of directions. The analysis of regularity of these general basis functions is a difficult task and we have shown that the basis functions obtained in the two-directional and unbalanced constructions are regular within cosets.

Non-linear approximation and compression of images

In Chapter 5, we have investigated the applications of directionlets in non-linear approximation and compression of images. We have shown that, allowing for spatial segmentation and choosing transform directions in each segment independently, directionlets outperform the standard 2-D wavelet transform in terms of asymptotic behavior of the mean-square error in non-linear approximation of C^2/C^2 images. Furthermore, we have demonstrated that directionlets can improve substantially the performance of the zerotree-based compression algorithms, where they substitute the standard 2-D wavelet transform.

Denoising of images

In Chapter 6, we have presented a novel denoising algorithm based on the undecimated version of directionlets. The method uses the Gaussian scale mixture model to estimate the transform coefficients from a noisy observation. The denoising results achieved by this method are comparable to the current state-of-the-art results, while having lower complexity because of the separable filtering operations.

7.2 Future Work

Sparse representation of geometrical features in images, as very important visual perception elements, has been a topic of high interest in the last decade. During the research work on directionlets, several questions have been arisen and left for future work due to the complexity of the problems.

On the other hand, directionlets provide a much wider framework for an efficient description of multi-dimensional signals. Thus, the signals like video sequences or sets of images or video sequences of the same physical scene recorded in separate cameras have high correlation among samples that can be exploited by appropriate methods. Investigation of the role of directionlets in these methods will be a topic of future work.

Thus, several topics are of interest as extension of this work and they are listed as follows.

Regularity of general multi-directional bases

Importance of regular basis functions in image compression has been emphasized in Section 4.3. However, it has been shown that neither cyclic nor unbalanced construction achieve regularity in general. It is of great interest to propose a construction method that can lead to regular orthogonal multi-channel filter banks and impose directional vanishing moments in the corresponding basis functions, while discarding the constraint of separable filtering. These truly 2-D filter-banks, although requiring substantially more complex design, would provide a powerful toolbox for an efficient image representation.

Compression and denoising of images

The compression methods that are based on optimal spatial-frequency quantization use the standard 2-D wavelet transform or wavelet packets. The general multi-directional wavelet transforms are convenient for the applications in these compression algorithms because of critical sampling. However, search for optimal values of a number of parameters in the compression method is a very challenging task, since the global optimization process has an exponential complexity. The goal of the future work is to propose an efficient ("smart")

optimization that still has a non-exponential complexity.

A more complex (non-quadtrees) spatial segmentation is capable of extracting more flexibly dominant directions in images than the isotropic segmentation. However, such segmentation methods are more costly in terms of encoding the corresponding segmentation tree. In the future work on the compression algorithm, we aim at an efficient trade-off between the representation sparsity and coding efficiency.

Truly 2-D multi-channel filter-banks that impose directional vanishing moments along more than two directions provide an efficient denoising tool. These filter-banks would be able to capture local coherence in images across space, directions, and scales better than the separable schemes. However, the 2-D filtering requires a higher computational complexity. The goal is to present an efficient trade-off between the properties of the filter-banks and complexity of the processing in the future work.

Video coding

The state-of-the-art video coding algorithms are based on motion compensation, and they are more efficient if the decorrelation processing applied within a frame or across frames provides a sparser representation. The goal is to propose an efficient video coding algorithm based on multi-resolution multi-directional anisotropic transforms that can capture geometrical features and motion (across space and time) in video sequences.

Distributed still image and video coding

Images of the same scene recorded from different spatial locations (multiple camera systems) are strongly correlated. The goal in distributed coding is to make an optimal use of the correlation structure across images while minimizing the amount of communication. Similarly, frames in a video sequence have high redundancy across space and time. This can be exploited by algorithms based on distributed principles (Wyner-Ziv) in which the transmission bitrate is reduced and the complexity is moved from the encoder to the decoder.

Besides the redundancy within one video sequence, two sequences of the same scene generated from two separated cameras at certain relative spatial distance and angle (stereo video) are also redundant. In addition, distributed coding techniques allow for an increase in resilience since the motion compensation takes place at the decoder.

Bibliography

- [1] D. Andrews and C. Mallows. Scale mixtures of normal distributions. *J. R. Statist. Soc.*, 36:99, 1974.
- [2] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Trans. Image Processing*, 1(2):205–220, April 1992.
- [3] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice Hall, Englewood Cliffs, NJ, 1982.
- [4] R. H. Bamberger and M. J. T. Smith. A filter bank for the directional decomposition of images: Theory and design. *IEEE Trans. Signal Processing*, pages 882–893, April 1992.
- [5] G. E. P. Box and C. Tiao. *Bayesian Inference in Statistical Analysis*. Addison-Wesley, Reading, MA, 1992.
- [6] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, CA, 1984.
- [7] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965.
- [8] W. Cai and M. Adjouadi. Minimization of boundary artifacts on scalable image compression using symmetric-extended wavelet transform. In *Proc. on the International Conference on Information Technology: Coding and Computing*, Las Vegas, NV, April 2004.
- [9] E. J. Candès and D. Donoho. Curvelets and curvilinear integrals. Technical report, Department of Statistics, Stanford University, CA, December 1999.
- [10] E. J. Candès and D. L. Donoho. Curvelets - a surprisingly effective nonadaptive representation for objects with edges. In A. Cohen, C. Rabut, and L. L. Schumaker, editors, *Curve and Surface Fitting*. Vanderbilt University Press, Saint-Malo, 1999.
- [11] E. J. Candès and D. L. Donoho. New tight frames of curvelets and optimal representations of objects with smooth singularities. Technical report, Department of Statistics, Stanford University, 2002.
- [12] A. Chambolle, R. A. DeVore, N. Lee, and B. J. Lucier. Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Trans. Image Processing*, 7(3):319–335, March 1998.

-
- [13] T. S. Chan and R. K. K. Yip. Line detection algorithm. In *Proc. IEEE International Conference on Pattern Recognition*, volume 2, pages 126–130, 1996.
- [14] S. G. Chang, B. Yu, and M. Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE Trans. Image Processing*, 9(9):1532–1546, September 2000.
- [15] S. G. Chang, B. Yu, and M. Vetterli. Spatially adaptive wavelet thresholding with context modeling for image denoising. *IEEE Trans. Image Processing*, 9(9):1522–1531, September 2000.
- [16] T. Chen and P. P. Vaidyanathan. Multidimensional multirate filters and filter banks derived from one-dimensional filters. *IEEE Trans. Signal Processing*, 41(5):1749–1765, May 1993.
- [17] A. Cohen and I. Daubechies. Non-separable bidimensional wavelet bases. *Revista Matematica Iberoamericana*, 9(1):51–137, 1993.
- [18] A. Cohen and I. Daubechies. A new technique to estimate the regularity of refinable functions. *Revista Matematica Iberoamericana*, 12(2):527–591, 1996.
- [19] A. Cohen and B. Matei. Compact representation of images by edge adapted multiscale transforms. In *Proc. IEEE International Conference on Image Processing (ICIP2001)*, Thessaloniki, Greece, October 2001.
- [20] R. R. Coifman and D. L. Donoho. Translation-invariant de-noising. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*. Springer-Verlag, Berlin, Germany, 1995.
- [21] J. H. Conway and N. J. A. Sloane. *Sphere packings, lattices and groups*. Springer, New York, NY, 1998.
- [22] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk. Wavelet-based statistical signal processing using hidden Markov models. *IEEE Trans. Signal Processing*, 46(4):886–902, April 1998.
- [23] A. L. Cunha, J. Zhou, and M. N. Do. The nonsubsampling contourlet transform: theory, design, and applications. *IEEE Trans. Image Processing*, submitted for publication, 2005.
- [24] A. L. da Cunha and M. N. Do. Bi-orthogonal filter banks with directional vanishing moments. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2005)*, pages 553–556, Philadelphia, PA, March 2005.
- [25] I. Daubechies. *Ten lectures on wavelets*. SIAM, Philadelphia, PA, 1992.
- [26] R. A. DeVore. Nonlinear approximation. *Acta Numer.*, 7:51–150, 1998.
- [27] M. N. Do. *Directional Multiresolution Image Representations*. PhD thesis, School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland, 2001.
- [28] M. N. Do and M. Vetterli. The contourlet transform: An efficient directional multiresolution image representation. *IEEE Trans. Image Processing*, to appear.

-
- [29] M. E. Domínguez-Jiménez and N. González Prelicic. Linear boundary extensions for finite length signals and paraunitary two-channel filterbanks. *IEEE Trans. Signal Processing*, 52(11):3213–3226, November 2004.
- [30] D. L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Inform. Theory*, 41(3):613–627, May 1995.
- [31] D. L. Donoho. Wedgelets: Nearly-minimax estimation of edges. *Ann. Statist.*, 27:859–897, 1999.
- [32] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, 81:425–455, 1994.
- [33] D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Assoc.*, 90(432):1200–1224, December 1995.
- [34] D. L. Donoho and I. M. Johnstone. Wavelet shrinkage: Asymptotia? *J. R. Stat. Soc., ser. B*, 57(2):301–369, 1995.
- [35] D. L. Donoho, M. Vetterli, R. A. DeVore, and I. Daubechies. Data compression and harmonic analysis. *IEEE Trans. Inform. Theory*, 44:2435–2476, October 1998.
- [36] P. L. Dragotti. *Wavelet Footprints and Frames for Signal Processing and Communications*. PhD thesis, School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland, 2002.
- [37] P. L. Dragotti and M. Vetterli. Footprints and edgeprints for image denoising and compression. In *Proc. IEEE International Conference on Image Processing (ICIP2001)*, Thessaloniki, Greece, October 2001.
- [38] P. L. Dragotti and M. Vetterli. Wavelet footprints: theory, algorithms and applications. *IEEE Trans. Signal Processing*, 51(5):1306–1323, May 2003.
- [39] M. Figueiredo and R. Nowak. Wavelet-based image estimation: An empirical bayes approach using jeffrey’s noninformative prior. *IEEE Trans. Image Processing*, 10(9):1322–1331, September 2001.
- [40] J. D. Foley, A. V. Dam, S. K. Feiner, and J. F. Hughes. *Computer graphics: Principles and Practice*. Adilson-Wesley Publishing Company, Reading, MA, 1990.
- [41] G. H. Granlund and H. Knutsson. Compact associative representation of visual information. In *Proc. on 10th International Conference on Pattern Recognition*, volume ii, 16-21, pages 200–207, June 1990.
- [42] C. Herley. Boundary filters for finite-length signals and time-varying filter banks. *IEEE Trans. Circuits Syst. II*, 42(2):102–114, February 1995.
- [43] C. Herley and M. Vetterli. Orthogonal time-varying filter banks and wavelet packets. *IEEE Trans. Signal Processing*, 42(10):2650–2663, October 1994.
- [44] K. Hirakawa and T. W. Parks. Adaptive homogeneity-directed demosaicing algorithm. *IEEE Trans. Image Processing*, 14(3):360–369, March 2005.

-
- [45] Y. Huang, I. Pollak, and C. A. Bouman. Image compression with multitree tilings. In *Proc. on (IEEE) International Conference on Acoustics, Speech, and Signal Proc.*, pages 193–196, March 2005.
- [46] Y. Huang, I. Pollak, M. N. Do, and C. A. Bouman. Optimal tilings and best basis search in large dictionaries. In *Conf. record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, pages 327–331, November 2003.
- [47] Y. Huang, I. Pollak, M. N. Do, and C. A. Bouman. Fast search for best representations in multitree dictionaries. *IEEE Trans. Image Processing*, to appear.
- [48] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall Inc., Englewood Cliffs, NJ, 1989.
- [49] I. M. Johnstone and B. W. Silverman. Wavelet threshold estimators for data with correlated noise. *J. R. Statist. Soc.*, 59, 1997.
- [50] N. Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Journal of Appl. and Comput. Harmonic Analysis*, 10:234–253, 2001.
- [51] J. Kovačević. *Filter Banks and Wavelets: Extensions and Applications*. PhD thesis, Graduate School of Arts and Sciences, Columbia Univ., New York, NY, 1991.
- [52] J. Kovačević and M. Vetterli. Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for \mathbb{R}^n . *IEEE Trans. Inform. Theory*, pages 533–555, March 1992.
- [53] R. Leonardi and M. Kunt. Adaptive split and merge for image analysis and coding. *Proc. SPIE*, 594, 1985.
- [54] X. Li and M. T. Orchard. Spatially adaptive image denoising under overcomplete expansion. In *Proc. IEEE International Conference on Image Processing (ICIP2000)*, pages 300–303, Vancouver, BC, Canada, September 2000.
- [55] X. Li and M. T. Orchard. New edge-directed interpolation. *IEEE Trans. Image Processing*, 10(10):1521–1527, October 2001.
- [56] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, CA, 1997.
- [57] F. G. Meyer and R. R. Coifman. Brushlets: a tool for directional image analysis and image compression. *Journal of Appl. and Comput. Harmonic Analysis*, 4:147–187, 1997.
- [58] J. Milner. *Mondrian*. Phaidon, London, UK, 2002.
- [59] T. K. Moon and W. C. Stirling. *Mathematical Methods and Algorithms for Signal Processing*. Prentice Hall, New Jersey, NJ, 2000.
- [60] D. D. Muresan and T. W. Parks. Prediction of image detail. In *Proc. IEEE International Conference on Image Processing (ICIP2000)*, pages 323–326, Vancouver, BC, Canada, September 2000.

-
- [61] D. D. Muresan and T. W. Parks. Adaptive optimal-recovery image interpolation. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2001)*, pages 1949–1952, Salt Lake City, UT, May 2001.
- [62] R. D. Nowak and R. G. Baraniuk. Wavelet-based transformations for nonlinear signal processing. *IEEE Trans. Signal Processing*, 47:1852–1865, July 1999.
- [63] T. Oka, T. Uto, and M. Ikehara. Smooth signal extension for m-channel paraunitary filterbanks and its application to image coding. In *Proc. on (IEEE) International Conference on Image Proc.*, volume 2, pages 229–232, September 2003.
- [64] E. L. Pennec and S. Mallat. Image compression with geometric wavelets. In *Proc. IEEE International Conference on Image Processing (ICIP2000)*, pages 661–664, Vancouver, Canada, September 2000.
- [65] E. Le Pennec and S. Mallat. Bandelet image approximation and compression. *SIAM MMS*, 4(3):992–1039, 2005.
- [66] E. Le Pennec and S. Mallat. Sparse geometric image representations with bandelets. *IEEE Trans. Image Processing*, 14:423–438, April 2005.
- [67] S.-M. Phoong, C. W. Kim, P. P. Vaidyanathan, and R. Ansari. A new class of two-channel biorthogonal filter banks and wavelet bases. *IEEE Trans. Signal Processing*, 43(3):649–665, March 1995.
- [68] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Trans. Image Processing*, 12(11):1338–1351, November 2003.
- [69] W. K. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1988.
- [70] K. Ramchandran. *Joint Optimization Techniques in Image and Video Coding with Applications to Multiresolution Digital Broadcast*. PhD thesis, Columbia University, New York, NY, 1993.
- [71] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Trans. Image Processing*, 32(2):160–175, April 1993.
- [72] O. Rioul. A discrete-time multiresolution theory. *IEEE Trans. Signal Processing*, 41(8):2591–2606, August 1993.
- [73] J. K. Romberg, H. Choi, and R. G. Baraniuk. Bayesian tree-structured image modeling using wavelet-domain hidden Markov models. *IEEE Trans. Image Processing*, 10(7):1056–1068, July 2001.
- [74] J. K. Romberg, M. Wakin, and R. Baraniuk. Multiscale wedgelet image analysis: fast decompositions and modeling. In *Proc. IEEE International Conference on Image Processing (ICIP2002)*, pages 585–588, Rochester, NY, September 2002.

-
- [75] J. K. Romberg, M. Wakin, and R. Baraniuk. Approximation and compression of piecewise smooth images using a wavelet/wedgelet geometric model. In *Proc. IEEE International Conference on Image Processing (ICIP2003)*, pages 49–52, Barcelona, Spain, September 2003.
- [76] C. Pe Rosiene and T. Q. Nguyen. Tensor-product wavelet vs. Mallat decomposition: A comparative analysis. In *Proc. IEEE International Symposium on Circuits and Systems (ISCAS1999)*, pages 431–434, Orlando, FL, June 1999.
- [77] J. G. Rosiles and M. J. T. Smith. A low complexity overcomplete directional image pyramid. In *Proc. IEEE International Conference on Image Processing (ICIP2003)*, Barcelona, Spain, September 2003.
- [78] F. Ruggeri and B. Vidaković. A bayesian decision theoretic approach to wavelet thresholding. *Statist. Sinica*, 9(1):183–197, 1999.
- [79] A. Said and W. A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits Syst. Video Technol.*, 6:243–250, June 1996.
- [80] I. W. Selesnick. Multiwavelet bases with extra approximation properties. *IEEE Trans. Signal Processing*, 46(11):2898–2908, November 1998.
- [81] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. Signal Processing*, 41:3445–3463, December 1993.
- [82] Y. Shoham and A. Gersho. Efficient bit allocation for an arbitrary set of quantizers. *IEEE Trans. Acoust., Speech, Signal Processing*, 36(9):1445–1453, September 1988.
- [83] R. Shukla. *Rate-distortion Optimized Geometrical Image Processing*. PhD thesis, School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland, 2004.
- [84] R. Shukla, P. L. Dragotti, M. N. Do, and M. Vetterli. Rate-distortion optimized tree-structured compression algorithms for piecewise polynomial images. *IEEE Trans. Image Processing*, 14:343–359, March 2005.
- [85] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger. Shiftable multiscale transforms. *IEEE Trans. Inform. Theory*, 38:587–607, March 1992.
- [86] A. Skodras, C. Christopoulos, and T. Ebrahimi. The JPEG 2000 still image compression standard. *IEEE Signal Processing Mag.*, 18:36–58, September 2001.
- [87] J.-L. Starck, E. J. Candès, and D. L. Donoho. The curvelet transform for image denoising. *IEEE Trans. Image Processing*, 11(6):670–684, June 2002.
- [88] G. Strang and V. Strela. Orthogonal multiwavelets with vanishing moments. *Opt. Eng.*, 33(7):2104–2107, 1994.
- [89] P. Strobach. Quadtree structured recursive plane decomposition coding of images. *IEEE Trans. Signal Processing*, 39(6):1380–1397, June 1991.

-
- [90] G. J. Sullivan and R. L. Baker. Efficient quadtree coding of images and video. *IEEE Trans. Image Processing*, 3(3):327–331, May 1994.
- [91] D. Taubman and M. Marcellin. *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001.
- [92] D. B. H. Tay and N. G. Kingsbury. Flexible design of multidimensional perfect reconstruction FIR 2-band filters using transformations of variables. *IEEE Trans. Image Processing*, 2(4):466–480, October 1993.
- [93] G. Tzagkarakis, B. Beferull-Lozano, and P. Tsakalides. Rotation-invariant texture retrieval with gaussianized steerable pyramids. *IEEE Trans. Image Processing*, to appear.
- [94] T. Uto and M. Ikehara. A smooth extension for the nonexpansive orthogonal wavelet decomposition of finite length signals. In *Proc. on (IEEE) International Conference on Acoustics, Speech, and Signal Proc.*, volume 3, pages 625–628, April 2003.
- [95] P. P. Vaidyanathan. *Multirate systems and filter banks*. Prentice Hall, Englewood Cliffs, NJ, 1993.
- [96] D. J. Vaisey and A. Gersho. Image coding with variable block size segmentation. *IEEE Trans. Signal Processing*, 40(8):2040–2060, August 1992.
- [97] V. Velisavljević, B. Beferull-Lozano, M. Vetterli, and P. L. Dragotti. Discrete multi-directional wavelet bases. In *Proc. IEEE International Conference on Image Processing (ICIP2003)*, pages 1025–1028, Barcelona, Spain, September 2003.
- [98] V. Velisavljević, B. Beferull-Lozano, M. Vetterli, and P. L. Dragotti. Approximation power of directionlets. In *Proc. IEEE International Conference on Image Processing (ICIP2005)*, Genova, Italy, September 2005.
- [99] V. Velisavljević, B. Beferull-Lozano, M. Vetterli, and P. L. Dragotti. Directionlets: Anisotropic multi-directional representation with separable filtering. *IEEE Trans. Image Processing*, to appear.
- [100] V. Velisavljević, P. L. Dragotti, and M. Vetterli. Directional wavelet transforms and frames. In *Proc. IEEE International Conference on Image Processing (ICIP2002)*, pages 589–592, Rochester, NY, September 2002.
- [101] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding*. Prentice Hall PTR, New Jersey, NJ, 1995.
- [102] M. Vetterli and G. Strang. Time-varying filter banks and multiwavelets. In *Sixth IEEE Digital Signal Processing Workshop*, pages 223–226, October 1994.
- [103] E. Viscito and J. P. Allebach. The analysis and design of multidimensional FIR perfect reconstruction filter banks for arbitrary sampling lattices. *IEEE Trans. Circuits Syst.*, pages 29–41, January 1991.
- [104] G. Wackersreuther. On two-dimensional polyphase filter banks. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-34(1):192–199, February 1986.

-
- [105] M. J. Wainwright and E. P. Simoncelli. Scale mixtures of gaussians and the statistics of natural images. In S. A. Solla, T. K. Leen, and K. R. Müller, editors, *Adv. Neural Information Processing Systems*. MIT Press, Cambridge, MA, 2000.
- [106] M. J. Wainwright, E. P. Simoncelli, and A. S. Willsky. Random cascades on wavelet trees and their use in modeling and analyzing natural imagery. *Appl. Comput. Harmon. Anal.*, 11(1):89–123, July 2001.
- [107] M. Wakin, J. Romberg, H. Choi, and R. Baraniuk. Wavelet-domain approximation and compression of piecewise smooth images. *IEEE Trans. Image Processing*, accepted.
- [108] M. Wakin, J. Romberg, C. Hyeokho, and R. Baraniuk. Rate-distortion optimized image compression using wedgelets. In *Proc. IEEE International Conference on Image Processing (ICIP2002)*, pages III–237 – III–240, Rochester, NY, September 2002.
- [109] A. B. Watson. The cortex transform: rapid computation of simulated neural images. *Computer Vision, Graphics, and Image Processing*, 39(3):311–327, 1987.
- [110] P. H. Westerink. *Subband Coding of Images*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 1989.
- [111] I. H. Witten, R. Neal, and J. G. Cleary. Arithmetic coding for data compression. *Comm. ACM*, 30:520–540, June 1987.
- [112] X. Wu. Image coding by adaptive tree structured segmentation. *IEEE Trans. Inform. Theory*, 38(6):1755–1767, November 1992.
- [113] Z. Xiong, K. Ramchandran, and M. T. Orchard. Space-frequency quantization for wavelet image coding. *IEEE Trans. Image Processing*, 6:677–693, May 1997.
- [114] Z. Xiong, K. Ramchandran, and M. T. Orchard. Wavelet packet image coding using space-frequency quantization. *IEEE Trans. Image Processing*, 7:892–898, June 1998.
- [115] C. Q. Zhan and L. J. Karam. Wavelet-based adaptive image denoising with edge preservation. In *Proc. IEEE International Conference on Image Processing (ICIP2003)*, pages 97–100, Barcelona, Spain, September 2003.
- [116] R. A. Zuidwijk. Directional and time-scale wavelet analysis. *SIAM Journal on Mathematical Analysis*, 31(2):416–430, 2000.

Curriculum Vitae

Vladan Velisavljević

Ecole Polytechnique Fédérale de Lausanne (EPFL)
School of Computer and Communication Sciences
Audiovisual Communications Laboratory (LCAV)
BC-320, Station 14, CH-1015 Lausanne, Switzerland
E-mail: vladan.velisavljevic@epfl.ch
Tel. +41 21 693 7663, Fax. +41 21 693 4312
<http://lcavwww.epfl.ch/~vladan>

Education

- 2001 – 2005 **Ph.D. Candidate**, Audiovisual Communications Laboratory (LCAV), School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland.
- 2001 **Doctoral School in Communications**, School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland.
- 2000 **M.S. in Electrical Engineering**, School of Electrical Engineering, University of Belgrade, Serbia, Yugoslavia.
- 1998 **Dipl.Ing. in Electrical Engineering**, School of Electrical Engineering, University of Belgrade, Serbia, Yugoslavia.

Professional Experience

- 2001-present **Research and Teaching Assistant**, Audiovisual Communications Laboratory (LCAV), School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland.
- Summer 2003 **Summer Research Internship**, Communication and Signal Processing Group, Department of Electrical and Electronic Engineering, Imperial College, London, UK.
- 1999 – 2000 **Research and Teaching Assistant**, Communications Department, University of Belgrade, Serbia, Yugoslavia.

Personal

Born in Sabac, Serbia, Yugoslavia, on October 22, 1974. Nationality: Serbian.

Publications

Journals:

[1] V. Velisavljević, B. Beferull-Lozano, M. Vetterli and P. L. Dragotti. **Directionlets: anisotropic multi-directional representation with separable filtering.** *IEEE Trans. on Image Proc.*, *accepted*, August 2005.

Conferences:

[2] V. Velisavljević, B. Beferull-Lozano, M. Vetterli and P. L. Dragotti. **Approximation power of directionlets.** *Proc. of the IEEE Int. Conf. on Image Processing (ICIP2005)*, Genova, Italy, September 2005.

[3] V. Velisavljević, B. Beferull-Lozano, M. Vetterli and P. L. Dragotti. **Discrete multi-directional wavelet bases.** *Proc. of the IEEE Int. Conf. on Image Processing (ICIP2003)*, Barcelona, Spain, September 2003.

[4] P. L. Dragotti, V. Velisavljević, M. Vetterli and B. Beferull-Lozano. **Discrete directional wavelet bases and frames for image compression and denoising.** *Proc. of the SPIE Conference on Wavelet Applications in Signal and Image Processing, X Wavelets*, San Diego, USA, August 2003, invited paper.

[5] P.L. Dragotti, V. Velisavljević, M. Vetterli and B. Beferull-Lozano. **Discrete directional wavelet bases for image compression.** *Proc. of the Visual Communications and Image Processing (VCIP)*, Lugano, Switzerland, July 2003, invited paper.

[6] V. Velisavljević, P. L. Dragotti, M. Vetterli. **Directional wavelet transforms and frames.** *Proc. of the IEEE Int. Conf. on Image Processing (ICIP2002)*, Rochester, USA, September 2002.

[7] P. L. Dragotti, M. Vetterli and V. Velisavljević. **Directional wavelets and wavelet footprints for compression and denoising.** *Proc. of the Int. Workshop of Advanced Methods for Multimedia Signal Processing (IWDC2002)*, Capri, Italy, September 2002, invited paper.

[8] V. Velisavljević and M. Popović. **Image compression using wavelet transform with various types of extension.** *Proc. of the Int. Workshop of Electrical Engineering (ETRAN2000)*, Soko Banja, Serbia, Yugoslavia, June 2000.

[9] V. Velisavljević and Z. Dobrosavljević. **Detection of DTMF Signals Using FFT.** *Proc. of the Int. Telecommunication Forum (TELFOR1997)*, Belgrade, Serbia, Yugoslavia, November 1997.

Others:

[10] V. Velisavljević. **Applications of the wavelet transform in image compression** (available only in Serbian). M.S. Dissertation. School of Electrical Engineering, University of Belgrade, September 2000.

THE LITTLE COUSIN SERIES IN MATHEMATICAL SIGNAL PROCESSING

Editor: Martin Vetterli

The Columbia series

1. Karlsson, Gunnar David. *Subband Coding for Packet Video*. CU/CTR/TR 137-89-16, May 1989.
2. Linzer, Elliot Neil. *Arithmetic Complexity and Numerical Properties of Algorithms involving Toeplitz Matrices*. October 1990.
3. Kovačević, Jelena. *Filter Banks and Wavelets: Extensions and Applications*. CU/CTR/TR 257-91-38, September 1991.
4. Uz, Kamil Metin. *Multiresolution Systems for Video Coding*. CU/CTR/TR 313-92-23, May 1992.
5. Radha, Hayder M. Sadik. *Efficient Image Representation using Binary Space Partitioning Trees*. CU/CTR/TR 343-93-23, December 1992.
6. Nguyen, Truong-Thao. *Deterministic Analysis of Oversampled A/D Conversion and Sigma/Delta Modulation, and Decoding Improvements using Consistent Estimates*. CU/CTR/TR 327-93-06, February 1993.
7. Herley, Cormac. *Wavelets and Filter Banks*. CU/CTR/TR 339-93-19, April 1993.
8. Garrett, Mark William. *Contributions toward Real-Time Services on Packet Switched Networks*. CU/CTR/TR 340-93-20, April 1993.
9. Ramchandran, Kannan. *Joint Optimization Techniques in Image and Video Coding with Applications to Multiresolution Digital Broadcast*. June 1993.
10. Shah, Imran Ali. *Theory, Design and Structures for Multidimensional Filter Banks and Applications in Coding of Interlaced Video*. CU/CTR/TR 367-94-14, December 1993.
11. Hong, Jonathan Jen-I. *Discrete Fourier, Hartley, and Cosine Transforms in Signal Processing*. CU/CTR/TR 366-94-13, December 1993.
12. Ortega, Antonio. *Optimization Techniques for Adaptive Quantization of Image and Video under Delay Constraints*. CU/CTR/TR 374-94-21, June 1994.

The Berkeley years

13. Park, Hyung-Ju. *A Computational Theory of Laurent Polynomial Rings and Multidimensional FIR Systems*. Coadv. with Tsit-Yuen Lam, Mathematics, U.C. Berkeley. UCB/ERL M95/39, May 1995.
14. Cvetković, Zoran. *Overcomplete Expansions for Digital Signal Processing*. UCB/ERL M95/114, December 1995.
15. McCanne, Steven Ray. *Scalable Compression and Transmission of Internet Multicast Video*. Coadv. with Van Jacobson, Lawrence Berkeley National Laboratory. UCB/CSD 96/928, December 1996.

16. Goodwin, Michael Mark. *Adaptive Signal Models: Theory, Algorithms, and Audio Applications*. Coadv. with Edward A. Lee, EECS, U.C. Berkeley. UCB/ERL M97/91, December 1997.
17. Goyal, Vivek K. *Beyond Traditional Transform Coding*. UCB/ERL M99/2, September 1998.
18. Chang, Sai-Hsueh Grace. *Image Denoising and Interpolation based on Compression and Edge Models*. Coadv. with Bin Yu, Statistics, U.C. Berkeley. UCB/ERL M99/57, Fall 1998.

The Lausanne time

19. Prandoni, Paolo. *Optimal Segmentation Techniques for Piecewise Stationary Signals*. EPFL 1993(1999), June 1999.
20. Lebrun, Jérôme. *Balancing MultiWavelets*. EPFL 2192(2000), May 2000.
21. Weidmann, Claudio. *Oligoquantization in Low-Rate Lossy Source Coding*. EPFL 2234(2000), July 2000.
22. Balmelli, Laurent. *Rate-Distortion Optimal Mesh Simplification for Communication*. EPFL 2260(2000), September 2000.
23. Marziliano, Pina. *Sampling Innovations*. EPFL 2369(2001), April 2001.
24. Horbelt, Stefan. *Splines and Wavelets for Image Warping and Projection*. EPFL 2397(2001), May 2001.
25. Hasler, David. *Perspectives on Panoramic Photography*. EPFL 2419(2001), July 2001.
26. Do, Minh N. *Directional Multiresolution Image Representations*. EPFL 2500(2001), November 2001.
27. Dragotti, Pier Luigi. *Wavelet Footprints and Frames for Signal Processing and Communications*. EPFL 2559(2002), April 2002.
28. Pecenovic, Zoran. *Integrating Visual and Semantic Descriptions for Effective, Flexible and User-Friendly Image Retrieval*. EPFL 2679(2002), October 2002.
29. Hu, Xiao-Yu. *Low-Delay Low-Complexity Error-Correcting Codes on Sparse Graphs*. EPFL 2681(2002), November 2002.
30. Gastpar, Michael. *To Code Or Not To Code*. EPFL 2687(2002), December 2002.
31. Polotti, Pietro. *Fractal Additive Synthesis: Spectral Modeling of Sound for Low-Rate Coding of High-Quality Audio*. EPFL 2711(2002), December 2002.
32. Cristescu, Răzvan. *Efficient Decentralized Communications in Sensor Networks*. EPFL 2952(2004), March 2004.
33. Shukla, Rahul. *Rate-Distortion Optimized Geometrical Image Processing*. EPFL 2992(2004), May 2004.

34. Viste, Harald. *Binaural Localization and Separation Techniques*. EPFL 3043(2004), June 2004.
35. Faller, Christof. *Parametric Coding of Spatial Audio*. EPFL 3062(2004), July 2004.
36. Maravić, Irena. *Sampling Methods for Parametric Non-Bandlimited Signals : Extensions and Applications*. EPFL 3058(2004), July 2004.
37. Ridolfi, Andrea. *Power Spectra of Random Spikes and Related Complex Signals, with Application to Communications*. EPFL 3157(2004), December 2004.