Directional Multiresolution Image Representations

Minh N. Do

October 23, 2001

Kính tặng Bố Mẹ

Abstract

Efficient representation of visual information lies at the foundation of many image processing tasks, including compression, filtering, and feature extraction. Efficiency of a representation refers to the ability to capture significant information of an object of interest in a small description. For practical applications, this representation has to be realized by structured transforms and fast algorithms. Recently, it has become evident that commonly used separable transforms (such as wavelets) are not necessarily best suited for images. Thus, there is a strong motivation to search for more powerful schemes that can capture the intrinsic geometrical structure of pictorial information.

This thesis focuses on the development of new "true" two-dimensional representations for images. The emphasis is on the discrete framework that can lead to algorithmic implementations. The first method constructs multiresolution, local and directional image expansions by using non-separable filter banks. This discrete transform is developed in connection with the continuous-space curvelet construction in harmonic analysis. As a result, the proposed transform provides an efficient representation for two-dimensional piecewise smooth signals that resemble images. The link between the developed filter banks and the continuous-space constructions is set up in a newly defined directional multiresolution analysis. The second method constructs a new family of block directional and orthonormal transforms based on the ridgelet idea, and thus offers an efficient representation for images that are smooth away from straight edges. Finally, directional multiresolution image representations are employed together with statistical modeling, leading to powerful texture models and successful image retrieval systems.

Résumé

La représentation adéquate de l'information visuelle est à la base des techniques de traitement du signal, de compression et d'identification de structure. Par représentation adéquate on entends la description succincte et précise d'une partie de la scène qui soit digne d'intérêt. En pratique, cette description doit être générée par des transformations structurelles et des algorithmes rapides. Récemment, il est apparut évident que les transformations usuelles de type séparables, telles que la transformée en ondelettes, ne représentent pas la panacée des outils de traitement de l'images. Le besoin de disposer d'outils permettant l'identification de la structure géométrique d'une scène se fait ainsi sentir.

Cette thèse met l'accent sur la recherche de nouvelles techniques permettant une description « rééllement bidimensionelle »de l'image, et plus particulièrement sur les méthodes discrètes qui conduisent à des algorithmes utilisables en pratique. La première technique construit une décomposition multirésolution, locale et directionnelle de l'image en utilisant un banc de filtre multidimensionnel. Cette décomposition est développée en analogie avec la récente transformation en « curvelets »utilisée dans le domaine de l'analyse harmonique, et fournit une description compacte des fonction bidimensionelles continues par morceau, qui sont considérées comme de bonnes représentations des images naturelles. Le lien entre les courbelettes et les bancs de filtre est démontrée dans un cadre nouvellement définit d'analyse multi-résolution directionnelle. La seconde technique présentée dans cette thèse construit une nouvelle famille de transformées directionnelle orthogonales en blocs basée sur l'idée des « ridgelets », qui offre une représentation adéquate des images qui sont régulières hors des zones de contours rectilignes. Finalement, la représentation directionnelle et multi-résolution des images est utilisée conjointement avec des méthodes statistiques permettant ainsi la création de modèles de textures performants ainsi que la création d'outils puissants de recherches d'images par contenu.

Contents

List of Figures

List of Tables

Acknowledgments

This thesis is the outcome of a wonderful three-year working experience under the guidance of Martin Vetterli, to whom I wish to express my appreciation and gratitude. His vision, creativeness, and enthusiasm have not just fueled this research but also have a lasting influence in my career as a role model.

I take this opportunity to thank my friends and colleagues at EPFL for their generous help and fruitful collaborations, especially Pier Luigi Dragotti, Michael Gastpar, David Hasler, Aurélie Lozano, Arrate Muñoz, Irena Maravić, Pina Marziliano, Zoran Pečenović, Jocelyne Plantefol, and Rahul Shukla. Email is a great asset for a young researcher to seek for suggestions and feedback from many experts. I am grateful to the discussions and interactions with Richard Baraniuk, Thierry Blu, Emmanuel Candès, David Donoho, Vivek Goyal, Gabor Herman, Jelena Kovačević, Stéphane Mallat, Ivan Selesnick, Eero Simoncelli, Mark Smith, and Michael Unser. This work was supported by a PhD Fellowship from the Department of Communication Systems, EPFL, which I gratefully acknowledge.

My deepest gratitude goes to my wife My and my son Trí. They have given me the much needed comfort and inspiration for completing this thesis. Finally, I would like to dedicate this work to my parents, whose constant support and encouragement have really brought me here.

Chapter 1

Introduction

1.1 Motivation

Over the last decade or so, wavelets have had a growing impact on many fields due to their unifying role in theory as well as their success in applications. At the heart of the theory, wavelets provide us with the construction of bases for signal expansion. Given a signal x, it can be represented by a linear combination of elementary signals called wavelets, $\{\psi_n\}_{n\in\mathbb{N}}$, as

$$x = \sum_{n=0}^{+\infty} c_n \psi_n, \quad \text{where} \quad c_n = \langle x, \psi_n \rangle. \tag{1.1}$$

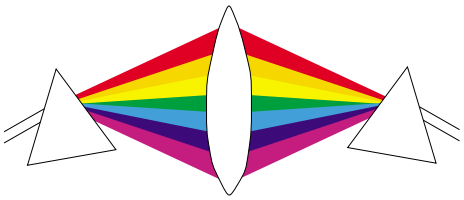


Figure 1.1: Light can be represented by a mixture of primary colors. This was first discovered by Newton in 1672 [116].

To visualize the concept, consider the following illustrative experiment shown in Figure 1.1. Assume that a ray of light source is *decomposed* into several color components. With a right mixture of those colors, we can *reproduce* the original light source at the other end. In this experiment, seven primary colors provide a *basis* for representing any light sources; in the same way as wavelets are bases for representing signals. Such decomposition is useful since it allows us to treat each component independently, in the same spirit as the "divide and conquer" strategy.

In many applications, one would like to *approximate* a signal given its basis expansion (1.1) [103]. One way to do that is to keep the *first-M* components,

for instance the first two colors: red and yellow, in the light source experiment. This is referred to as *linear approximation* since it is equivalent to projecting the input object onto the subspace spanned by the first-M basis elements. The linear approximation can be written as

$$\hat{x}_M^{(LA)} = \sum_{n=0}^{M-1} c_n \psi_n. \tag{1.2}$$

However, if a green-dominant light source comes in then the linear approximation would produce a very poor result. A better scheme for approximation is to keep the *best-M* components in the color expansion. This is referred to as *non-linear approximation* since it is adaptive and is based on the input signal [46, 47]. In general, the non-linear approximation can be written as

$$\hat{x}_M^{(NLA)} = \sum_{n \in I_M} c_n \psi_n, \tag{1.3}$$

where I_M is the set of M indices corresponding to the M-largest $|c_n|$. The non-linear approximation scheme certainly works well for the green-dominant light source, and in fact for any light source that is made up of few colors.

As a more applied example, consider the approximation of a one-dimensional *piecewise smooth* signal with $N = 1024$ samples using both linear and non-linear approximation schemes [173]. In both cases, only $M = 64$ coefficients are kept. Figure 1.2 shows the results using the Fourier basis. In this case we see that the non-linear approximation is not necessarily better than linear approximation, but both provide poor results. Moreover, there are annoying artifacts, due to Gibbs oscillations [118], in the reconstructed signals.

Figure 1.3 shows the results of the same experiment but using a wavelet basis. With the linear approximation scheme, the resulting signal is also very poor. On the other hand, the result using non-linear approximation with wavelets is excellent: the signal is almost perfectly recovered with, we reemphasize, 64 out of 1024 wavelet coefficients.

We can therefore consider the "efficiency" of an expansion: that is, for most *objects of interest*, the expansion is *sparse*, or can be well represented by a few components. Clearly, this efficient representation is realized by a non-linear approximation scheme. We observe that efficiency is the main objective of many representation systems. For the human visual system, efficiency means that few visual neurons need to be excited when we see a typical view. For image compression or content-based retrieval systems, the use of efficient representation implies the compactness of the compressed file or the index entry for a typical image in the database. The experiments with the 1-D signal above indicate that *wavelets provide a very **sparse** or **efficient** representation for piecewise smooth signals*. This is the key reason for the success of wavelets in many signal processing applications.

Given the success of wavelets for 1-D piecewise smooth signals, one might asks "is this the end of the story?". Unfortunately, commonly used separable wavelet transforms (that is obtained by applying a 1-D transform separately in

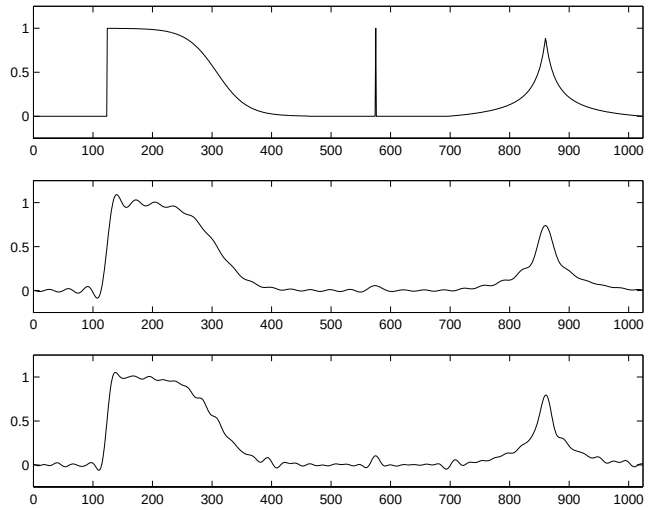


Figure 1.2: Approximation using Fourier basis. The original signal has $N = 1024$ samples. The approximated signals are reconstructed from $M = 64$ Fourier coefficients that are either the *first* coefficients (linear approximation) or the *most significant* coefficients (non-linear approximation). Note that the peak in the middle of the original signal is completely missed in both schemes. E is measured as the sum of sample square errors.

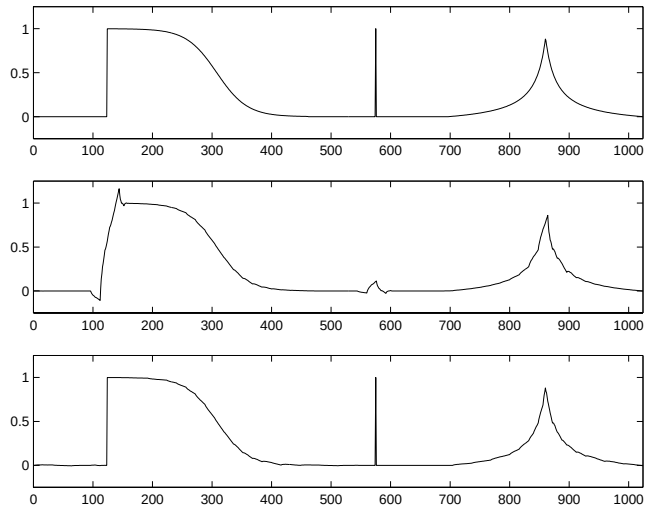
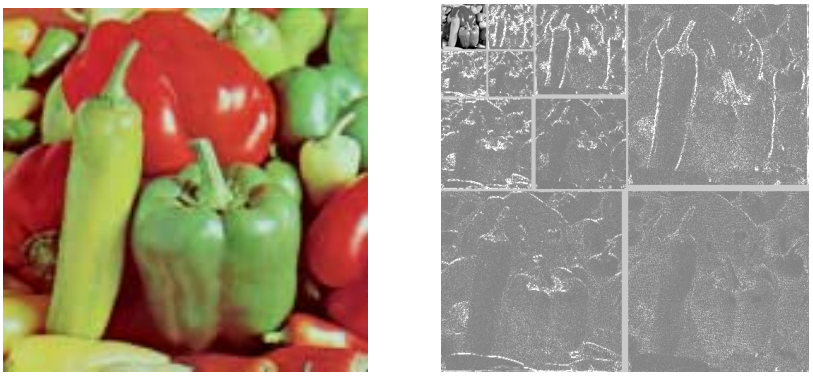


Figure 1.3: Same experiment as in Figure 1.2, but using a wavelet basis with two vanishing moments and six levels of decomposition. The signal is almost perfectly recovered in the non-linear approximation scheme.

each dimension) have some limitations in higher dimensions. Figure 1.4 shows an example of the 2-D wavelet transform of a natural image. The significant

wavelet coefficients are shown in white dots, which are around the points with sharp contrasts in the intensity – or edges of the image. The key point to note here is that the locations of these significant coefficients exhibit *visible geometrical correlation* as they form simple curves. In essence, wavelets in 2-D are good at catching *edge points*, but do not see the smoothness along *contours*. This disappointing behavior indicates that more powerful representations are needed in higher dimensions.



(a) (b)

Figure 1.4: Wavelet transform of an image. (a) The "peppers" image. (b) The positions of the significant wavelet coefficients are shown in white, they are around the edges of the image.

To be more precise, to first order, we can consider a simple model for natural images that is composed of piecewise smooth regions separated by smooth boundaries, as shown in Figure 1.5. In computer vision, it is also known that boundaries of shapes provide the most useful information. This model obviously ignores the important image information, namely texture, and we will address the texture modeling problem later on. In the 2-D piecewise smooth model, the discontinuities are generated by *edges* – referred to as points in the image where there is a sharp contrast in the intensity – whereas edges are often gathered along smooth *contours* that are created by typically smooth boundaries of physical objects. With this image model, the goal is to explore the *intrinsic geometrical structure* of natural images. In short, the action is at the edges!

To visualize the limitation of the 2-D separable wavelet transform in representing images with smooth contours, we consider the following scenario. Imagine that two painters, one with a "wavelet"-style and the other with a "X-let"-style, want to paint a natural scene (Figure 1.6). Both painters apply a refinement technique to increase resolution from coarse to fine. In this case, efficiency is measured by how quickly, that is with the least number of brush strokes, one can faithfully reproduce the scene. Consider the situation when a

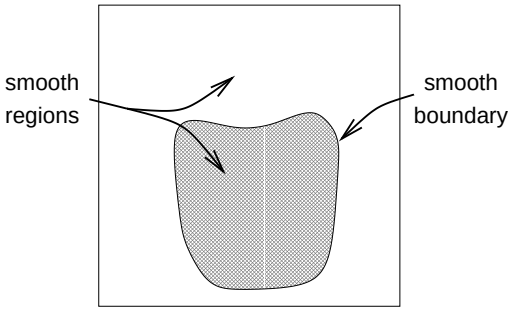


Figure 1.5: A model for natural images that is composed of piecewise smooth regions separated by smooth boundaries.

smooth contour is being painted. As 2-D wavelets are constructed from tensor products of 1-D wavelets, the "wavelet"-style painter is limited to use brushes of square shapes along the contour, with different sizes corresponding to the multiresolution of wavelets. As the resolution is getting finer we can clearly see the limitation of the wavelet scheme since it requires many finer "dots" to capture the contour. The "X-let"-style, on the other hand, has more freedom in making brush strokes in different directions and rectangular shapes that follow the contour. It can be clearly seen that the "X-let"-style is much more efficient than the "wavelet"-style. In fact, "X-let" is the underlying spirit of the recent *curvelet* system [20] in harmonic analysis that will be studied and implemented in this thesis.

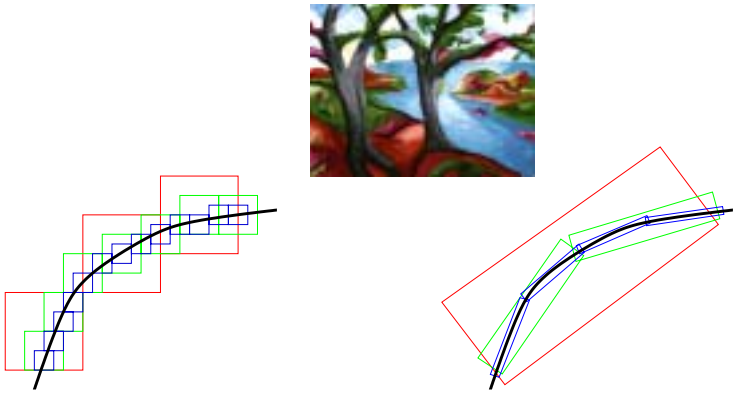


Figure 1.6: Wavelet versus new scheme: illustrating the successive refinement of a smooth contour by the two systems.

The main point to note from this painting scenario is that a separable wavelet scheme with isometric basis elements fails to recognize that the boundary is smooth. To be more efficient, new schemes have to be richer in directions and support shapes for their basis elements.

We can even look further to see what we can learn from nature. Visual information is very well captured and represented by the human visual system. Psychology studies show that only 20 bits per second is what can be taken in by the human eye [70]. Yet with that rate, it typically takes only a few seconds or about 100 bits to capture a visual scene. By contrast, to represent a typical image of size 512×512 pixels with the most advanced image compression system, namely JPEG-2000 which uses wavelets, at least tens of kilobytes are required! This suggests there is still plenty of room for improvements in the current techniques for image representation.

What makes the human visual system very efficient in representing pictorial information? We know from physiological studies [82, 44] that the receptive fields in the visual cortex are characterized as being *localized*, *oriented* and *bandpass*. Recently, several studies to identify the sparse components of natural image patches of small sizes [117]; produced results that are shown in Figure 1.7. Strikingly, these sparse components resemble closely the aforementioned characteristics of the visual cortex. This matching points to the efficiency hypothesis of the human visual system that its purpose is to capture the essential information with a minimum number of excited neurons. Without this, our brain would have to use much more energy when we see the world surrounding us! Furthermore, we observe a distinguishing feature of the sparse codes in Figure 1.7, namely that they can be organized into a *pyramidal directional structure*.

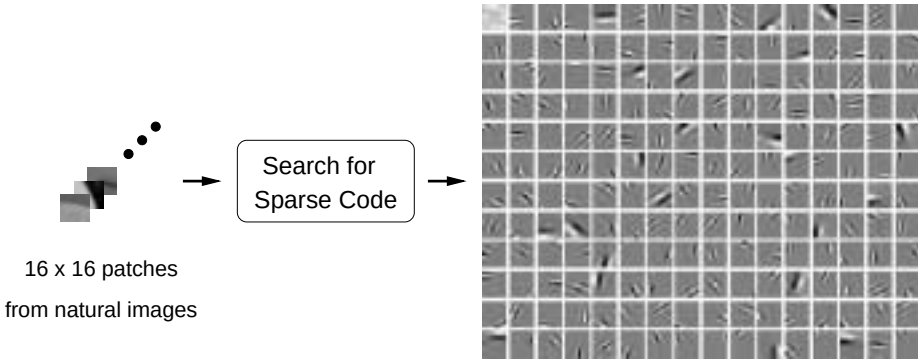


Figure 1.7: Experiment producing sparse components of natural image by Olshausen and Field [117]. The 16×16 image patches extracted from natural scenes are used to search for a basis that maximizes both preserve information and sparseness of the expansions.

1.2 Problem Statement

The discussion from the previous section presents a strong motivation for "true" 2-D representation schemes that can facilitate the next generation of image processing applications. The primary goal of the thesis is to develop structured transforms with fast algorithms that provide sparse representations for two-

dimensional piecewise smooth signals resembling images. Our emphasis is on a discrete framework that can lead to algorithmic implementations. Inspired by the painting scenario and the studies related to the human visual system, in the following we identify a "wish list" for new image representations:

Multiresolution: The representation should allow images to be successively approximated starting from a coarse version and going to fine-resolution version.

Localization: The basis elements in the representation should be well concentrated in both spatial and the frequency domains.

Critical sampling: For some applications (e.g. compression), the representation should be a basis, or a frame with small redundancy.

Directionality: The representation should contain basis functions oriented at variety of directions, much more than the three directions that are offered by separable wavelets.

Anisotropy: To capture smooth contours in images, the representation should contain basis functions with variety of shapes, in particular with different aspect ratios.

Among these desiderata, the first three are successfully provided by the separable wavelet system. However, the last two require new challenging *non-separable* constructions.

Efficient or sparse image expansion is only the first step in many image processing applications. The second objective of the thesis is to explore multiscale directional decompositions in visual information retrieval application, especially for texture images. Here, the key issue is on the modeling of the transform coefficients, together with correct similarity measurements between models.

To sum up, in this thesis we aim to:

- Develop new image representations with the aforementioned properties on a discrete framework that can lead to algorithmic implementations.
- Derive an analysis framework that precisely connect the newly developed discrete transforms with the continuous-domain constructions.
- Explore sparse image representations together with statistical modeling for image retrieval and other applications.

1.3 Related Work

Perhaps one of the most interesting result in wavelet research is the connection between the wavelet transform in harmonic analysis and filter banks in discrete signal processing (Figure 1.8). The connection was set up by the multiresolution analysis by Mallat and Meyer [101, 112]. Such a connection allows the wavelet

transform, initially defined in the continuous domain, to be computed with fast algorithms based on filter banks. The multiresolution analysis provides a natural framework for the understanding of wavelet bases, and starting from iterated filter banks leads to the famous construction of compactly supported wavelets by Daubechies [40].

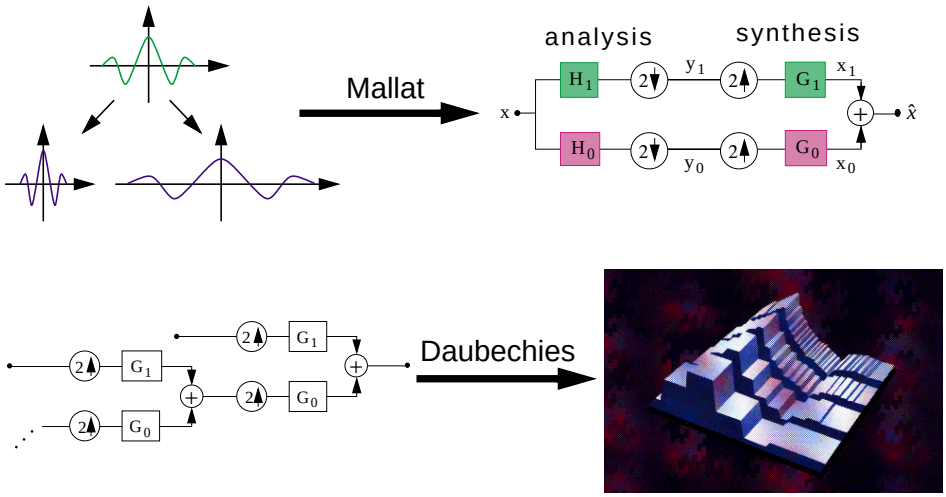


Figure 1.8: Fundamental links between wavelets and filter banks. The three dimensional plot is taken from the cover of [175], which represents the convergence of a discrete filter to a continuous scaling function.

From the filter bank perspective, there are two schemes that can lead to construction of two-dimensional wavelet bases [169, 175]. A straightforward method is to use *separable* filter banks as mentioned before, which leads to tensor product wavelet (Figure 1.9).

An alternative is to use *non-separable* filter banks, with the most famous example being the quincunx filter bank [93, 30] (Figure 1.10). Non-separable could be referred to either the filters or the sampling lattice. These filter banks typically have higher complexity and the filter design issue is much more challenging as compared to the 1-D case.

Nevertheless, as can be seen from the frequency partition diagrams, both the separable and non-separable quincunx filter banks are very limited in treating directionality. In 1992, Bamberger and Smith [9] explored the non-separable feature in constructing a 2-D directional filter bank that can be maximally decimated while achieving perfect reconstruction. At that time, the focus was on the directional decomposition for image analysis purposes, thus the issues related to the construction of bases for image representation and the link with the continuous domain were not addressed.

An interesting and important question that relates the filter banks with the construction of bases is what conditions should be imposed on the filters of an iterated filter bank so that iterations lead to regular functions? The situation

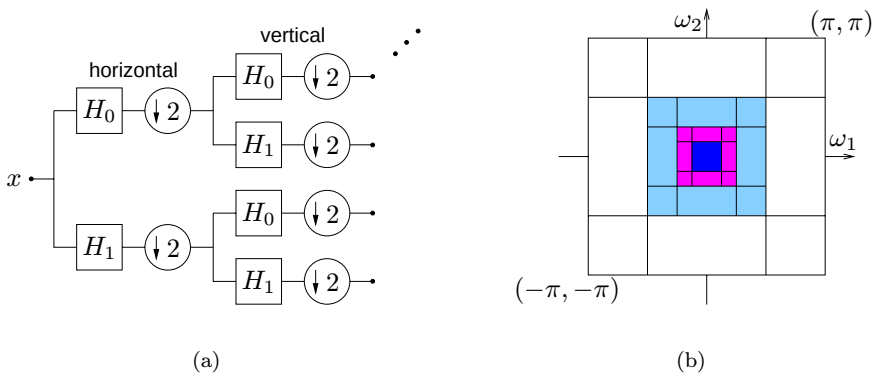


Figure 1.9: Separable wavelet filter bank in 2-D. (a) Block diagram. The filters are in one-dimensional. The iteration is done on the "low-low" band. (b) Frequency partition.

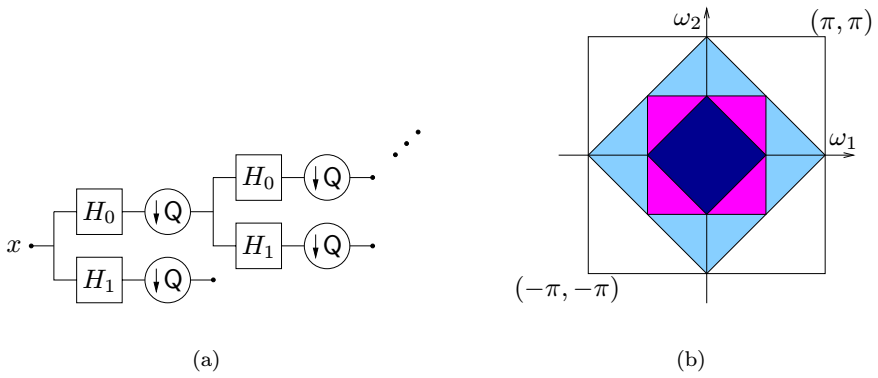


Figure 1.10: A non-separable wavelet filter bank: the quincunx case. (a) Block diagram. The filters are two-dimensional. Sampling is represented by sub-lattices. The iteration is done on the lowpass band. (b) Frequency partition.

in 1-D is illustrated in Figure 1.11. Understanding this regularity condition is the key in the construction of wavelets from filter banks [42, 160]. In general, the regularity issue for 2-D iterated filter banks is much more involved than in the 1-D case (see for example [33]).

Several well-know systems that provide multiscale and directional representations include the cortex transform [178], the steerable pyramid [152], two-dimensional directional wavelets [4], brushlets [111] and complex wavelets [90]. The main difference between these systems with the one developed in this thesis is that the previous methods do *not* either aim at critical sampling or connect to continuous-domain constructions through iterated filter banks.

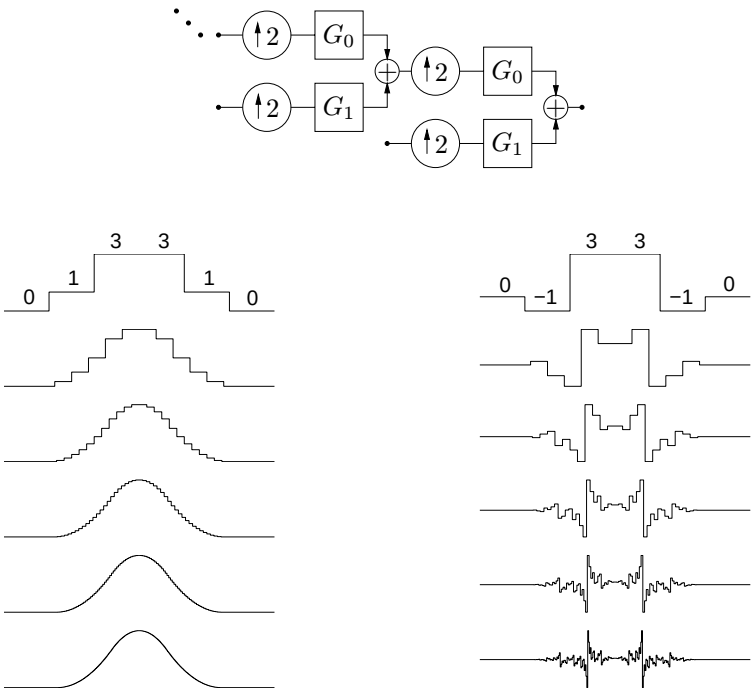


Figure 1.11: Iterated 1-D filter bank where two similar filters lead to very different iteration behaviors.

A key insight into the construction of directional bases from the Radon transform was provided by Candès and Donoho [19, 21] with the *ridgelet transform*. The idea is to map a line singularity into a point singularity using the Radon transform. Then, the wavelet transform can be used to handle the point singularity. The result is an efficient representation for 2-D functions with piecewise smooth regions separated by a line singularity.

The ridgelet transform is then extended by the same authors into the *curvelet transform* [20] that can handle singularities along smooth curves. The idea is that by smooth windowing, a curve singularity is broken into a line singularity and can be handled by a local ridgelet transform. The curvelet construction lead to the surprising fact that efficient representation of 2-D piecewise smooth functions can be achieved with a *fixed* transform that has a correct handling of space, scale and direction. The initial ridgelet and curvelet transforms are both defined in continuous spaces.

A different approach to handle the geometrical regularity of images is taken by the recent *bandelet transform* [125]. In this transform, the discontinuity curves are detected before hand and explicitly represented by an adaptive expansion.

1.4 Thesis Outline

The thesis is organized as follows. In the next three chapters, we focus on the constructions of bases and frames for image representation using multi-dimensional filter banks. In **Chapter 2**, the Laplacian pyramid (LP) introduced by Burt and Adelson [18] is studied as a frame operator, and this reveals that the usual reconstruction is sub-optimal. With orthogonal filters, the LP is shown to be a tight frame, and thus the optimal linear reconstruction using the dual frame operator has a simple structure which is symmetrical with the forward transform. In more general cases, we propose an efficient filter bank (FB) for the reconstruction of the LP which uses a projector and performs better than the usual method. Next, by setting up the LP as an oversampled FB, we give a complete parameterization of all synthesis FB's providing perfect reconstruction. Finally, we consider the situation where the LP scheme is iterated and derive the continuous-time frame associated with the LP.

Chapter 3 studies a two-dimensional directional filter bank (DFB) proposed by Bamberger and Smith [9] that can be maximally decimated while achieving perfect reconstruction. We develop it further, in particular under the angle of providing orthogonal bases. To facilitate this, we provide a detail study on the behavior of the iterated DFB that includes an explicit formulation of the equivalent filters and the sampling matrices for the DFB channels. The properties of directional bases from the DFB is demonstrated with the "sinc" case, showing the existence of "local directional bases".

Chapter 4 proposes a new flexible multiscale and directional image representation, named *pyramidal directional filter banks*. The scheme combines the previous studied Laplacian pyramids and the directional filter banks to provide a sparse representation for two-dimensional piecewise smooth signals resembling images. The underlying expansion is a frame and can be designed to be a tight frame. Pyramidal directional filter banks can provide an effective method to implement the digital curvelet transform. The link between curvelets and pyramidal directional filter banks is established formally via a multiresolution framework, in much the same way as the continuous wavelets were connected with filter banks in one-dimension.

Initially, the ridgelet transform was defined in a continuous space. For practical applications, the development of discrete versions of the ridgelet transform that lead to algorithmic implementations is a challenging problem. In **Chapter 5**, we propose a version of the ridgelet transform for discrete and finite-size images. It uses the finite Radon transform introduced by Bolker [15] as a basic building block. The proposed transform is invertible, non-redundant and computed via fast algorithms. Furthermore, this construction leads to a large family of directional orthonormal bases for images. Numerical results show that the new transform is more effective than the wavelet transform in approximating and denoising images with straight edges.

An emerging multimedia application of directional multiresolution image representations, namely content-based image retrieval, is studied in **Chapter 6**.

First, we develop a general statistical framework for image retrieval by jointly considering the two problems of feature extraction and similarity measurement in a coherent manner, while keeping in mind the complexity constraint for such applications. This framework is then applied successfully to the texture retrieval problem, where it exhibits significant improvement in retrieval rate over existing methods. For this, we consider several wavelet-domain texture models, first by simply modeling marginal distributions of subband coefficients by generalized Gaussian densities, and later by the use of more complex hidden Markov models [38] that can capture the dependencies of wavelet descriptors across scales and orientations.

Finally, we conclude in **Chapter 7** with a summary and an outlook on future research.

Chapter 2

Framing Pyramid

2.1 Introduction and Motivation

Multiscale data representation is a powerful idea. It captures data in a hierarchical manner where each level corresponds to a reduced-resolution approximation. One of the early examples of such a scheme is the Laplacian pyramid (LP), proposed by Burt and Adelson [18] for image coding. The basic idea of the LP is the following. First, derive a coarse approximation of the original signal, by lowpass filtering and downsampling. Based on this coarse version, predict the original (by upsampling and filtering) and then calculate the difference as the prediction error. Usually, for reconstruction, the signal is obtained by simply adding back the difference to the prediction from the coarse signal. The process can be iterated on the coarse version. Analysis and usual synthesis of the LP are shown in Figure 2.1(a) and 2.1(b), respectively.

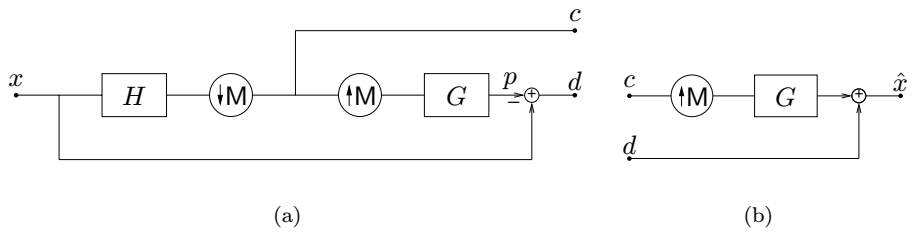


Figure 2.1: Laplacian pyramid scheme. (a) Analysis: the outputs are a coarse approximation c and a difference d between the original signal and the prediction. The process can be iterated by decomposing the coarse version repeatedly. (b) Usual synthesis.

A drawback of the LP is implicit oversampling. Therefore, in compression applications it is normally replaced by subband coding or wavelet transform

[0]This chapter includes research conducted jointly with Martin Vetterli [56, 57].

which is a critically sampled scheme and often an orthogonal decomposition. However, the LP has the advantage over the critically sampled wavelet scheme that each pyramid level generates only one bandpass signal, even for multidimensional cases. This property makes it easy to apply many multiresolution algorithms using a coarse-to-fine strategy [141] to the LP. Furthermore, the resulting bandpass signals of the LP do not suffer from the "scrambled" frequencies as in the critical sampling scheme. The reason for this frequency scrambling effect is illustrated in Figure 2.2 for the 1-D case. As can be seen, the highpass channel is folded back into the low frequency after downsampling, and thus its spectrum is reflected. In the LP, this effect is avoided by downsampling the lowpass channel only. Therefore the LP permits further subband decomposition to be applied on its bandpass images. A possible scheme is a pyramidal decomposition where the bandpass images of the LP are fed into directional filter banks [9]. The final result is a set of directional subband images at multiple scales [58].

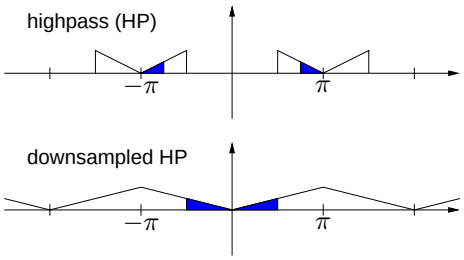


Figure 2.2: Illustration of the "frequency scrambling" in 1-D due to downsampling of the highpass channel. *Upper*: spectrum after highpass filtering. *Lower*: spectrum after downsampling. The filled regions indicate that the high frequency is folded back into the low frequency.

For many applications like compression and denoising, the coefficients in the transform domain are processed further, and this can introduce errors due to quantization or thresholding. The processed coefficients are then used to reconstruct the original data. For the LP, the usual reconstruction algorithm – adding the prediction from the coarse version with the difference, produces a perfectly reconstructed signal in the absence of noise, but turns out to be usually suboptimal otherwise.

Because the LP is an overcomplete representation (there are more coefficients after the analysis than in the input), it must be treated as a *frame* operator. Frames are generalizations of bases which lead to redundant expansions. A key observation is that one should use the *dual frame* operator for the reconstruction. While this seems a somewhat trivial observation, it has not been used in practice, probably because the usual reconstruction, while suboptimal, is very simple. Yet, we will show that gains around 1 dB are actually possible over the usual reconstruction. Beyond this improvement, we also believe that a full treatment of what is one of the most standard image processing structure is probably

worthwhile in its own right, be it only for pedagogical reasons.

Recently, there have been a considerable interest in wavelet and filter bank frames [127, 43, 29, 146, 92], where the greater design freedom available by the overcomplete systems leads to a variety of attractive features over bases. The LP frame offers a simple scheme that has low computation complexity (there is only one filtering channel), can be easily extended to higher dimensions (e.g. via separable extension), and has small redundancy (even in higher dimensions).

The outline of the chapter is as follows. In Section 2.2, we introduce the notation and set up the operators for the LP using both the time-domain and the polyphase-domain representations. The dual frame operator or the pseudo inverse is defined together with its properties for reconstruction. In Section 2.3, we consider the LP with orthogonal filters and show that it is a *tight frame*, and thus the pseudo inverse has a simple structure which is symmetrical with the forward transform. In Section 2.4, inspired by the structure of the pseudo inverse for the tight frame LP, we propose a new reconstruction for more general LP's that leads to better performance compared to the usual method. In Section 2.5, by setting up the LP as an oversampled filter bank (FB), we find a parameterization for all synthesis FB's providing perfect reconstruction for the LP. The oversampled FB view of the LP leads to a study of iterated LP and its associated continuous frames, so called framelets, in Section 2.6. We conclude in Section 2.7 with some discussions.

2.2 Preliminaries

2.2.1 Signals, Operators and Notations

Since the LP is valid for signals in any dimension and often used for images, we use multi-dimensional notation for generality. But throughout the chapter, formulas can also be simply regarded as one-dimensional. A discrete-time d-dimensional signal is a sequence of real-valued numbers defined on the integer lattice $\mathbb{Z}^d$, e.g. $x[n],\ n \in \mathbb{Z}^d$. Signals with finite energy belong to a Hilbert space $l_2(\mathbb{Z}^d)$ with the inner product defined as $\langle x, y\rangle = \sum_{n\in\mathbb{Z}^d} x[n]y[n]$, and thus the l_2-norm is $\|x\| = \sqrt{\langle x, x\rangle} = \sqrt{\sum_{n\in\mathbb{Z}^d} x[n]^2}$.

The z-transform of a discrete-time signal is denoted by

$$X(z) = \mathcal{Z}\{x[n]\} \stackrel{\triangle}{=} \sum_{n\in\mathbb{Z}^d} x[n]z^{-n}.$$

where raising a d-dimensional complex vector $z = (z_1, \ldots, z_d)^T$ to a d-dimensional integer vector $n = (n_1, \ldots, n_d)^T$ yields $z^n = \prod_{i=1}^d z_i^{n_i}$. On the unit hyper-sphere $z = e^{j\omega} \stackrel{\triangle}{=} (e^{j\omega_1}, \ldots, e^{j\omega_d})^T$, $X(e^{jw})$ is the Fourier transform of $x[n]$. For a matrix in the z-domain with real coefficients $\mathsf{A}(z)$, we will see that the equivalence of the transpose of a real matrix is $[\mathsf{A}(z^{-1})]^T$, which we denote by $\mathsf{A}^*(z)$. This notation is justified since on the unit hyper-sphere, $\mathsf{A}^*(e^{j\omega})$ is the transpose conjugation of $\mathsf{A}(e^{j\omega})$.

A filter is represented by its impulse response $h[n]$ or transfer function $H(z) = \mathcal{Z}\{h[n]\}$.[1] The output of the filter is the convolution of the input with the impulse response $h[n]$, i.e. $y[n] = x[n] * h[n] = \sum_{k\in\mathbb{Z}^d} x[k]h[n-k]$, or equivalently, $Y(z) = H(z)X(z)$.

The sampling operation is represented by a $d \times d$ nonsingular integer matrix M [169]. For an M-fold downsampling, the input $x[n]$ and the output $x_d[n]$ are related by

$$x_d[n] = x[\mathsf{M}n].$$

For an M-fold upsampling, the input $x[n]$ and the output $x_u[n]$ are related by

$$x_u[n] = \begin{cases} x[k] & \text{if } n = \mathsf{M}k,\ k \in \mathbb{Z}^d \\ 0 & \text{otherwise.} \end{cases}$$

In the z-domain, the upsampling operation can be simply written as $X_u(z) = X(z^{\mathsf{M}})$. The quantity z^{M} is defined to be a vector whose k-th element is given by z^{m_k}, where m_k is the k-th column of the matrix M. We denote $|\det(\mathsf{M})|$ by $|\mathsf{M}|$.

2.2.2 Burt and Adelson's Laplacian Pyramid

The structure of the Laplacian pyramid is shown in Figure 2.1. We concentrate first on one level of the LP; multilevel LP's are discussed later. The filtering and downsampling operation for the LP shown in Figure 2.1(a) yields the coarse approximation signal

$$c[n] = \sum_{k\in\mathbb{Z}^d} x[k]h[\mathsf{M}n - k] = \langle x, \tilde{h}[\cdot - \mathsf{M}n]\rangle, \tag{2.1}$$

where we denote $\tilde{h}[n] = h[-n]$. The upsampling and filtering operation results in

$$p[n] = \sum_{k\in\mathbb{Z}^d} c[k]g[n - \mathsf{M}k] = \sum_{k\in\mathbb{Z}^d} \langle x, \tilde{h}[\cdot - \mathsf{M}k]\rangle g[n - \mathsf{M}k]. \tag{2.2}$$

Writing signals as column vectors, for example $\boldsymbol{x} = (x[n] : n \in \mathbb{Z}^d)^T$, we can express these operations as left matrix multiplications

$$\boldsymbol{c} = \mathsf{H}\boldsymbol{x} \quad \text{and} \quad \boldsymbol{p} = \mathsf{G}\boldsymbol{c},$$

where H and G correspond to $(\downarrow \mathsf{M})H$ and $G(\uparrow \mathsf{M})$, respectively. For example,

[1] In the sequel, we often assume the filter to have finite impulse response (FIR) and thus $H(z)$ is polynomial. The results, however, also hold for infinite impulse response (IIR) filters by involving Laurent polynomials.

when $d = 1$ (1-D case) and $\mathsf{M} = 2$, we have

$$\mathsf{H} = \begin{pmatrix} \ddots & & & & & & & \\ \dots & h[2] & h[1] & h[0] & \dots & & & \\ & & \dots & h[2] & h[1] & h[0] & \dots & \\ & & & & & & \ddots & \end{pmatrix}, \text{ and}$$

$$\mathsf{G} = \begin{pmatrix} \ddots & \vdots & & \\ & g[0] & & \\ & g[1] & \vdots & \\ & g[2] & g[0] & \\ & \vdots & g[1] & \\ & & g[2] & \\ & & \vdots & \ddots \end{pmatrix}. \tag{2.3}$$

In general, H has $\left\{\tilde{h}[n - \mathsf{M}k]\right\}_{n\in\mathbb{Z}^d}$ as its rows and G has $\{g[n - \mathsf{M}k]\}_{n\in\mathbb{Z}^d}$ as its columns. Typically, those are infinite matrices but they can also be considered as finite matrices when dealing with finite length signals with appropriate boundary treatments. In the sequel, we denote I as the identity matrices with appropriate sizes depending on the context. Using this matrix notation, the difference signal of the LP can be written as

$$\boldsymbol{d} = \boldsymbol{x} - \boldsymbol{p} = \boldsymbol{x} - \mathsf{GH}\boldsymbol{x} = (\mathsf{I} - \mathsf{GH})\boldsymbol{x}.$$

By combining the previous relations, we can write the analysis operator of the LP as follows

$$\underbrace{\begin{pmatrix} \boldsymbol{c} \\ \boldsymbol{d} \end{pmatrix}}_{\boldsymbol{y}} = \underbrace{\begin{pmatrix} \mathsf{H} \\ \mathsf{I} - \mathsf{GH} \end{pmatrix}}_{\mathsf{A}} \boldsymbol{x}. \tag{2.4}$$

The usual inverse transform of the LP (refer to Figure 2.1(b)) computes: $\hat{\boldsymbol{x}} = \mathsf{G}\boldsymbol{c} + \boldsymbol{d}$, thus it can be written as

$$\hat{\boldsymbol{x}} = \underbrace{\begin{pmatrix} \mathsf{G} & \mathsf{I} \end{pmatrix}}_{\mathsf{S}_1} \begin{pmatrix} \boldsymbol{c} \\ \boldsymbol{d} \end{pmatrix}. \tag{2.5}$$

It is easy to check that $\mathsf{S}_1\mathsf{A} = \mathsf{I}$ for any H and G, which agrees with the well-known fact that the LP can be perfectly reconstructed with any pair of filters H and G.

As mentioned in the introduction, a drawback of the LP is its oversampling. After one step in the LP, we have a coarse version which is $1/|\mathsf{M}|$ times the size of the input and a difference signal that has the same size as the input. When the scheme is iterated we have the following redundancy ratio:

$$1 + \frac{1}{|\mathsf{M}|} + \frac{1}{|\mathsf{M}|^2} + \dots \to \frac{|\mathsf{M}|}{|\mathsf{M}| - 1}.$$

For example, in 2-D with $\mathsf{M} = 2 \cdot \mathsf{I}$, the oversampling is up to 33%. The redundancy of the LP goes down in higher dimensions and/or larger downsampling rates.

2.2.3 Polyphase-Domain Analysis

The matrix notation for the LP operations in the previous section is simple to use but it does not reveal the matrix block structure of the LP operators as can be seen in (2.3). To overcome this, we introduce the *polyphase-domain* representation [169, 175] for the LP. The polyphase decomposition of a signal with respect to the sampling matrix M is a set of $|\mathsf{M}|$ sub-signals which have the same indices modulo M, for example,

$$x_i[n] = x[\mathsf{M}n + k_i], \quad i = 0, 1, \ldots, |\mathsf{M}| - 1, \tag{2.6}$$

where $\{k_i\}_{0 \le i < |\mathsf{M}|}$ is the set of integer vectors of the form $\mathsf{M}\boldsymbol{t}$, such that $\boldsymbol{t} \in [0,1)^d$ [169]. The signal is reconstructed from its polyphase components by simply adding up the upsampled and shifted signals from those polyphase components. More precisely, we can write x in the z-domain as

$$X(z) = \sum_{k=0}^{|\mathsf{M}|-1} z^{-k_i} X_i(z^{\mathsf{M}}). \tag{2.7}$$

Therefore, a signal can be represented by the vector of its polyphase components, that is $\boldsymbol{x}(z) \stackrel{\triangle}{=} (X_0(z), \ldots, X_{|\mathsf{M}|-1}(z))^T$. The synthesis filter G is decomposed just as the signal, while the analysis filter H has reverse phase. With this representation, the output of the filtering and downsampling operation (2.1) can be written as

$$\begin{aligned} c[n] &= \sum_{i=0}^{|\mathsf{M}|-1} \sum_{m \in \mathbb{Z}^d} x[\mathsf{M}m + k_i] h[\mathsf{M}n - \mathsf{M}m - k_i] \\ &= \sum_{i=0}^{|\mathsf{M}|-1} \sum_{m \in \mathbb{Z}^d} x_i[m] h_i[n - m]. \end{aligned}$$

Thus taking the z-transform we have

$$C(z) = \sum_{i=0}^{|\mathsf{M}|-1} H_i(z) X_i(z) = \mathsf{H}(z)\boldsymbol{x}(z), \tag{2.8}$$

where $\mathsf{H}(z) \stackrel{\triangle}{=} (H_0(z), \ldots, H_{|\mathsf{M}|-1}(z))$. Similarly, the polyphase components of the upsampling and filtering operation (2.2) are

$$p_i[m] = p[\mathsf{M}m + k_i] = \sum_{n \in \mathbb{Z}^d} c[n] g[\mathsf{M}m + k_i - \mathsf{M}n] = \sum_{n \in \mathbb{Z}^d} c[n] g_i[m - n].$$

In z-domain, this can be written as

$$P_i(z) = G_i(z) C(z) \quad \text{or} \quad \boldsymbol{p}(z) = \mathsf{G}(z) C(z), \tag{2.9}$$

where $\mathsf{G}(z) \triangleq (G_0(z), \ldots, G_{|\mathsf{M}|-1}(z))^T$. Therefore, the polyphase vector of the difference signal in the LP is

$$\boldsymbol{d}(z) = \boldsymbol{x}(z) - \boldsymbol{p}(z) = (\mathsf{I} - \mathsf{G}(z)\mathsf{H}(z))\, \boldsymbol{x}(z).$$

So the analysis operator of the LP can be written in the polyphase-domain as

$$\underbrace{\begin{pmatrix} C(z) \\ \boldsymbol{d}(z) \end{pmatrix}}_{\boldsymbol{y}(z)} = \underbrace{\begin{pmatrix} \mathsf{H}(z) \\ \mathsf{I} - \mathsf{G}(z)\mathsf{H}(z) \end{pmatrix}}_{\mathsf{A}(z)} \boldsymbol{x}(z). \tag{2.10}$$

This clearly resembles the time-domain representation of the LP analysis operation in (2.4). Therefore in the sequel we can use the time-domain and the polyphase-domain representations interchangeably. Expressions derived for the time-domain representation also hold for the polyphase-domain representations with the obvious modifications and vice versa. Note that $\mathsf{A}(z)$ is a polynomial matrix of size $(|\mathsf{M}| + 1) \times |\mathsf{M}|$.

From this, it is straightforward that the usual inverse operator for the LP has the polyphase representation as

$$\hat{\boldsymbol{x}}(z) = \mathsf{S}_1(z)\, \boldsymbol{y}(z),$$

where

$$\mathsf{S}_1(z) = \begin{pmatrix} \mathsf{G}(z) & \mathsf{I} \end{pmatrix}.$$

2.2.4 Frame Analysis

Frames were first introduced by Duffin and Schaeffer [68]; for a detailed introduction, see [41, 42, 2, 103]. A family of functions $\{\phi_k\}_{k\in\Gamma}$ in a Hilbert space H is called a *frame* if there exist two constants $A > 0$ and $B < \infty$ such that

$$A\|f\|^2 \leq \sum_{k\in\Gamma} |\langle f, \phi_k\rangle|^2 \leq B\|f\|^2, \quad \forall f \in H, \tag{2.11}$$

where A and B are called the frame bounds. When $A = B$, the frame is said to be *tight*. Associated with a frame is the *frame operator* F, defined as the linear operator from H to $l_2(\Gamma)$ as

$$(Ff)_k = \langle f, \phi_k\rangle, \quad \text{for } k \in \Gamma. \tag{2.12}$$

It can be shown that the frame condition (2.11) is satisfied if and only if F is invertible on its range with a bounded inverse [103]. For the Laplacian pyramid, there always exists a bounded reconstruction inverse, which is the usual reconstruction, thus we immediately get the following result.

Proposition 2.1 *The LP with stable filters*[2] *provides a frame expansion in* $l_2(\mathbb{Z}^d)$.

[2]Stability of a filter means that a bounded input produces a bounded output.

As shown above, the frame operator for the LP is represented by a left matrix multiplication with A. Since the LP is a redundant transform, its frame operator admits an infinite number of left inverses. Among those, the most important is the *dual* frame operator or the *pseudo inverse* of A [80]

$$\mathsf{A}^{\dagger} = (\mathsf{A}^T\mathsf{A})^{-1}\mathsf{A}^T. \tag{2.13}$$

In the polyphase-domain, the pseudo inverse of $\mathsf{A}(z)$ is given by [39]

$$\mathsf{A}^{\dagger}(z) = (\mathsf{A}^*(z)\mathsf{A}(z))^{-1}\mathsf{A}^*(z). \tag{2.14}$$

2.2.5 Noise Analysis

When there is additive "noise" in the frame coefficients, the pseudo inverse eliminates the influence of errors that are orthogonal to the range of the frame operator. Therefore, if we have access to $\hat{\boldsymbol{y}} = \boldsymbol{y} + \boldsymbol{\eta}$ instead of $\boldsymbol{y} = \mathsf{A}\boldsymbol{x}$, then the pseudo inverse provides the solution $\hat{\boldsymbol{x}} = \mathsf{A}^{\dagger}\hat{\boldsymbol{y}}$ that minimizes the residual $\|\mathsf{A}\hat{\boldsymbol{x}} - \hat{\boldsymbol{y}}\|$. This is called the least-squares solution. For a tight frame, the pseudo inverse is simply the scaled transposed matrix of the frame operator, since $\mathsf{A}^T\mathsf{A} = A \cdot \mathsf{I}$.

We will now review results that allow us to quantify the performance of a left inverse. It can be shown [103] that the pseudo inverse has minimum *sup norm* among all the left inverses of the frame operator. Let S be an arbitrary left inverse of A. The sup norm of an operator S is defined as

$$\|\mathsf{S}\| = \sup_{\boldsymbol{y}\neq\mathbf{0}} \frac{\|\mathsf{S}\boldsymbol{y}\|}{\|\boldsymbol{y}\|}, \tag{2.15}$$

and for a matrix, it can be computed by [80]

$$\|\mathsf{S}\| = \max\{\sqrt{\lambda} : \lambda \text{ is an eigenvalue of } \mathsf{S}\mathsf{S}^T\}. \tag{2.16}$$

The influence of the sup norm in the reconstruction can be seen in the following. With the noise model setup as above, the reconstruction error by S is

$$\boldsymbol{\epsilon} \stackrel{\triangle}{=} \hat{\boldsymbol{x}} - \boldsymbol{x} = \mathsf{S}\hat{\boldsymbol{y}} - \boldsymbol{x} = \mathsf{S}(\boldsymbol{y} + \boldsymbol{\eta}) - \boldsymbol{x} = \mathsf{S}(\mathsf{A}\boldsymbol{x} + \boldsymbol{\eta}) - \boldsymbol{x} = \mathsf{S}\boldsymbol{\eta}. \tag{2.17}$$

Therefore,

$$\|\boldsymbol{\epsilon}\| \leq \|\mathsf{S}\|\|\boldsymbol{\eta}\|. \tag{2.18}$$

In other words, when the energy of the noise $\boldsymbol{\eta}$ is bounded, the sup norm of the inverse matrix provides an upper bound for the reconstruction error, and this bound is tight.

In some cases, we can assume that the additive noise $\boldsymbol{\eta}$ is white, which means that

$$E\{\eta[i], \eta[j]\} = \delta[i-j]\sigma^2, \quad \text{for all} \quad i, j, \tag{2.19}$$

or its autocorrelation matrix $\mathsf{R}_{\boldsymbol{\eta}} \triangleq E\{\boldsymbol{\eta}\boldsymbol{\eta}^T\} = \sigma^2 \cdot \mathsf{I}$. This noise model is approximately true when, for instance, $\boldsymbol{y}$ is uniformly scalar quantized. In this case, the autocorrelation of the reconstruction error by S is

$$\mathsf{R}_{\boldsymbol{\epsilon}} \triangleq E\{\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T\} = E\{\mathsf{S}\boldsymbol{\eta}\boldsymbol{\eta}^T\mathsf{S}^T\} = \mathsf{S}\mathsf{R}_{\boldsymbol{\eta}}\mathsf{S}^T = \sigma^2\mathsf{S}\mathsf{S}^T.$$

Hence, for signals of finite length N, the reconstruction mean squared error (MSE) is

$$MSE \triangleq N^{-1}E\{\|\boldsymbol{\epsilon}\|^2\} = N^{-1}\mathrm{tr}(\mathsf{R}_{\boldsymbol{\epsilon}}) = N^{-1}\sigma^2\mathrm{tr}(\mathsf{S}\mathsf{S}^T) \tag{2.20}$$

For infinite length signals that have polyphase representation defined as before, the reconstruction MSE can be computed in the Fourier domain using the power spectrum of $\boldsymbol{\epsilon}$, which is given by [169]

$$\mathsf{R}_{\boldsymbol{\epsilon}}(e^{j\omega}) = \mathsf{S}(e^{j\omega})\mathsf{R}_{\boldsymbol{\eta}}(e^{j\omega})\mathsf{S}^*(e^{j\omega}).$$

Therefore, similarly to (2.20), with the white noise model given in (2.19) we have

$$MSE = \frac{\sigma^2}{|\mathsf{M}|(2\pi)^d}\int_{[-\pi,\pi]^d}\mathrm{tr}(\mathsf{S}(e^{j\omega})\mathsf{S}^*(e^{j\omega}))d\omega. \tag{2.21}$$

Since the trace of a matrix equals to the sum of its eigenvalues, the eigenvalues of $\mathsf{S}\mathsf{S}^T$ and $\mathsf{S}(e^{j\omega})\mathsf{S}^*(e^{j\omega})$ (which are also the squares of the singular values of S and $\mathsf{S}(e^{j\omega})$, respectively) play an important role in analyzing the reconstruction error due to S. Using the orthogonal projection property of the pseudo inverse, it can be shown [42, 13] that among all the left inverses, the pseudo inverse minimize the reconstruction MSE due to white noise in the frame coefficients. In summary, the pseudo inverse provides the optimal linear reconstruction.

Example 2.1 *To get a gist of the aforementioned properties of frames, consider the following illustrative example. Suppose we have a redundant transform that takes a scalar $x \in \mathbb{R}$ and outputs a vector $\boldsymbol{y} = (y_1, y_2)^T \in \mathbb{R}^2$ such that $y_i = x,\ i = 1, 2$. There are infinite ways to reconstruct x from $\boldsymbol{y}$: one simple way is to assign $\hat{x}_1 = y_1$, another way is to compute $\hat{x}_2 = (y_1 + y_2)/2$. Under the white noise model given in (2.19), the performance by these two reconstruction methods can be quantified as: $MSE_1 = E\{\|x - \hat{x}_1\|^2\} = E\{\eta_1^2\} = \sigma^2$, and $MSE_2 = E\{\|x - \hat{x}_2\|^2\} = \frac{1}{4}E\{(\eta_1 + \eta_2)^2\} = \sigma^2/2$. Thus, we reduce the MSE by half by using the second reconstruction method instead of the first one. In fact, the second reconstruction method is the pseudo inverse, which minimizes the MSE in this case.*

2.3 Laplacian Pyramid with Orthogonal Filters

2.3.1 The Tight Frame Case

Consider a special case where the filters in the LP are orthogonal filters with respect to the sampling matrix M, which means

$$\begin{aligned} \langle g, g[\cdot - \mathsf{M}n] \rangle &= \delta[n], \quad \text{and} && (2.22) \\ h[n] &= g[-n], && (2.23) \end{aligned}$$

or equivalently, $\mathsf{G}^T\mathsf{G} = \mathsf{I}$ and $\mathsf{H} = \mathsf{G}^T$. The condition (2.23) can be written in the polyphase-domain as $\mathsf{H}^*(z) = \mathsf{G}(z)$. Since

$$\begin{aligned} \langle g, g[\cdot - \mathsf{M}n] \rangle &= \sum_{i=0}^{|\mathsf{M}|-1} \sum_{m \in \mathbb{Z}^d} g[\mathsf{M}m + k_i] g[\mathsf{M}m - \mathsf{M}n + k_i] \\ &= \sum_{i=0}^{|\mathsf{M}|-1} \sum_{m \in \mathbb{Z}^d} g_i[m] g_i[m-n], \end{aligned}$$

taking the z-transform of both sides of (2.22), the orthogonality condition on the filter G is equivalent to

$$\sum_{i=0}^{|\mathsf{M}|-1} G_i(z) G_i(z^{-1}) = 1 \quad \text{or} \quad \mathsf{G}^*(z)\mathsf{G}(z) = 1. \tag{2.24}$$

Orthogonal filters can be designed by using well-known methods, such as separable filters from 1-D orthogonal filters [169, 175] or non-separable filters in 2-D for $\mathsf{M} = 2 \cdot \mathsf{I}_2$ [87, 94].

Theorem 2.1 *The Laplacian pyramid with orthogonal filters is a* tight frame *with frame bounds equal to* 1. *Conversely, suppose the LP is a tight frame, then the frame bounds must be* 1 *and either* $H(z) = 0$ *or* $\mathsf{H}^*(z) = 2[\mathsf{G}^*(z)\mathsf{G}(z) + 1]^{-1}\mathsf{G}(z)$. *If the LP is a tight frame with FIR filters, and excluding the degenerated case* $H(z) = 0$, *then* $\mathsf{G}^*(z)\mathsf{G}(z) = 2\alpha - 1$ *and* $\mathsf{H}^*(z) = \alpha\mathsf{G}(z)$, *where* $\alpha \in \mathbb{R}$, *which means the filters are orthogonal up to a scale factor.*

Proof: Suppose that the Laplacian pyramid uses orthogonal filters. Using $\mathsf{G}^*(z)\mathsf{G}(z) = 1$ and $\mathsf{H}(z) = \mathsf{G}^*(z)$ we can directly verify that

$$\mathsf{A}^*(z)\mathsf{A}(z) = \begin{pmatrix} \mathsf{H}^*(z) & \mathsf{I} - \mathsf{H}^*(z)\mathsf{G}^*(z) \end{pmatrix} \begin{pmatrix} \mathsf{H}(z) \\ \mathsf{I} - \mathsf{G}(z)\mathsf{H}(z) \end{pmatrix} = \mathsf{I}.$$

Now suppose that LP is a tight frame, or $\|\boldsymbol{c}\|^2 + \|\boldsymbol{d}\|^2 = A\|\boldsymbol{x}\|^2$ for all $\boldsymbol{x} \in l_2(\mathbb{Z}^d)$. Since H is a decimating operator, there exists an $\boldsymbol{x} \neq \boldsymbol{0}$ such that $\mathsf{H}\boldsymbol{x} = \boldsymbol{0}$. In this case, the output of the LP is $\boldsymbol{c} = \boldsymbol{0}$, $\boldsymbol{d} = \boldsymbol{x}$. Hence the tight frame bound A must be 1.

So the tight frame condition for the LP becomes $\mathsf{A}^*(z)\mathsf{A}(z) = \mathsf{I}$. Expanding and grouping terms, this equation becomes

$$\mathsf{H}^*(z)[\mathsf{H}(z) - \mathsf{G}^*(z) + \mathsf{G}^*(z)\mathsf{G}(z)\mathsf{H}(z)] = \mathsf{G}(z)\mathsf{H}(z). \tag{2.25}$$

Let $\mathsf{K}(z) = \mathsf{H}(z) - \mathsf{G}^*(z) + \mathsf{G}^*(z)\mathsf{G}(z)\mathsf{H}(z)$. Note that $\mathsf{K}(z)$ is an $1 \times |\mathsf{M}|$ matrix, and denote $\mathsf{K}(z) = (K_0(z), \ldots, K_{|\mathsf{M}|-1}(z))$. The key observation is that both sides of (2.25) are *outer products* of two column vectors of $|\mathsf{M}|$ components. Thus, (2.25) can be written as

$$K_i(z)\mathsf{H}^*(z) = H_i(z)\mathsf{G}(z), \quad \text{for } \ i = 0, \ldots, |\mathsf{M}| - 1.$$

If $H(z) = 0$, then (2.25) holds and we have a degenerated LP tight frame since $\boldsymbol{c} = \mathbf{0}$, $\boldsymbol{d} = \boldsymbol{x}$ for all $\boldsymbol{x}$. Otherwise, there exists a polyphase component $H_i(z) \neq 0$, which implies $\mathsf{G}(z) = [K_i(z)/H_i(z)]\mathsf{H}^*(z) = \alpha(z)\mathsf{H}^*(z)$. Substitute this back to (2.25), it becomes

$$\alpha(z)^2\mathsf{H}^*(z)\mathsf{H}(z)\mathsf{H}^*(z)\mathsf{H}(z) = [2\alpha(z) - 1]\mathsf{H}^*(z)\mathsf{H}(z). \tag{2.26}$$

Since $H(z) \neq 0$, it can be shown that $\mathsf{H}(z)$ has right inverse $\bar{\mathsf{H}}(z)$, such that $\mathsf{H}(z)\bar{\mathsf{H}}(z) = 1$. Multiplying both sides of (2.26) with $\bar{\mathsf{H}}^*(z)$ on the left and $\bar{\mathsf{H}}(z)$ on the right, it is equivalent to $\mathsf{H}(z)\mathsf{H}^*(z) = [2\alpha(z) - 1]/\alpha(z)^2$. It follows that $\mathsf{G}^*(z)\mathsf{G}(z) = 2\alpha(z) - 1$, and thus $\mathsf{H}^*(z) = \alpha(z)^{-1}\mathsf{G}(z) = 2[\mathsf{G}^*(z)\mathsf{G}(z) + 1]^{-1}\mathsf{G}(z)$.

If in addition, we requires the filters to be FIR, then $\mathsf{G}^*(z)\mathsf{G}(z) = 2\alpha(z) - 1$ and $\mathsf{H}(z)\mathsf{G}(z) = 2 - 1/\alpha(z)$ are (finite) polynomials, which implies that $\alpha(z)$ must be a scalar. It follows that $H(z)$ and $G(z)$ are orthogonal filters. □

Using the well-known result on the non-existence of orthogonal two-channel filter banks with symmetric FIR filters (except for the Haar filters), we obtain the following corollary of the above theorem.

Corollary 2.1 *There are no LP tight frames for* $\mathsf{M} = 2$ *with symmetric FIR filters (except for* $H(z) = 0$ *or the Haar filters).*

Remark 2.1 *Under the orthogonality conditions, a geometrical interpretation of the tight frame can be seen by rewriting (2.2) as*

$$p[n] = \sum_{k \in \mathbb{Z}^d} \langle x, g[\cdot - \mathsf{M}k] \rangle g[n - \mathsf{M}k].$$

Denote V the subspace spanned by the set of orthogonal vectors $\{g[\cdot - \mathsf{M}k]\}_{k \in \mathbb{Z}^d}$. Then $\boldsymbol{p}$ is the orthogonal projection of $\boldsymbol{x}$ onto V. Together with the fact that $\boldsymbol{d} = \boldsymbol{x} - \boldsymbol{p}$, using the Pythagorean *theorem leads to*

$$\|\boldsymbol{x}\|^2 = \|\boldsymbol{p}\|^2 + \|\boldsymbol{d}\|^2 = \|\boldsymbol{c}\|^2 + \|\boldsymbol{d}\|^2. \tag{2.27}$$

where the equality $\|\boldsymbol{p}\| = \|\boldsymbol{c}\|$ comes from the fact that $\boldsymbol{c}$ are the coefficients in the orthonormal expansion (2.2) of $\boldsymbol{p}$ in V. Equation (2.27) says that the LP is a tight frame with the frame bounds $A = B = 1$.

Consequently, with orthogonal filters, the pseudo inverse of A is simply its transpose matrix. Thus,

$$\mathsf{A}^\dagger = \mathsf{A}^T = \begin{pmatrix} \mathsf{H} \\ \mathsf{I} - \mathsf{G}\mathsf{G}^T \end{pmatrix}^T = \begin{pmatrix} \mathsf{G} & \mathsf{I} - \mathsf{G}\mathsf{G}^T \end{pmatrix}.$$

Therefore the reconstruction using the pseudo inverse is

$$\hat{\boldsymbol{x}} = \mathsf{A}^{\dagger}\boldsymbol{y} = \mathsf{G}\boldsymbol{c} + (\mathsf{I} - \mathsf{G}\mathsf{G}^{T})\boldsymbol{d} = \mathsf{G}(\boldsymbol{c} - \mathsf{H}\boldsymbol{d}) + \boldsymbol{d}. \tag{2.28}$$

The last expression is derived in order to reduce the computational complexity of the pseudo inverse. It leads to an efficient filter bank structure for reconstruction of the LP that is shown in Figure 2.3. In the polyphase domain, the pseudo inverse becomes

$$\mathsf{A}^{\dagger}(z) = \mathsf{A}^{*}(z) = \begin{pmatrix} \mathsf{G}(z) & \mathsf{I} - \mathsf{G}(z)\mathsf{G}^{*}(z) \end{pmatrix}.$$

We emphasize two important facts here. First, the usual inverse is different from the pseudo inverse, and thus is suboptimal. We will make a precise comparison latter. Secondly, the pseudo inverse in this case has a symmetrical structure with the forward transform, and thus has the same order of complexity.

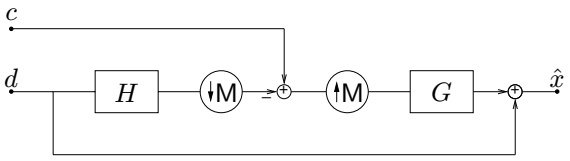


Figure 2.3: The proposed reconstruction scheme for the Laplacian pyramid in Figure 2.1(a). It is the pseudo inverse when the filters are orthogonal.

Example 2.2 (The Haar case) *Consider the 1-D LP with Haar filters:* $\tilde{h} = g = \left(1/\sqrt{2}, 1/\sqrt{2}\right)$ *and* $\mathsf{M} = 2$. *The output of the LP can be written as*

$$\begin{array}{rcrcl} c[n] & = & \frac{1}{\sqrt{2}}x[2n] & + & \frac{1}{\sqrt{2}}x[2n+1], \\ d[2n] & = & \frac{1}{2}x[2n] & - & \frac{1}{2}x[2n+1], \\ d[2n+1] & = & -\frac{1}{2}x[2n] & + & \frac{1}{2}x[2n+1]. \end{array} \tag{2.29}$$

In order to see the block structure of the transform matrices, we slightly change the notation and write $\boldsymbol{y} = (\ldots, c[n], d[2n], d[2n+1], \ldots)^{T}$. *Using (2.29), the analysis matrix of the LP is a block diagonal matrix*

$$\mathsf{A} = \begin{pmatrix} \ddots & & & \\ & \mathsf{B}_{\mathsf{A}} & & \\ & & \mathsf{B}_{\mathsf{A}} & \\ & & & \ddots \end{pmatrix},$$

where

$$\mathsf{B}_{\mathsf{A}} = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/2 & -1/2 \\ -1/2 & 1/2 \end{pmatrix}.$$

Similarly, the usual synthesis matrix S_1 *is also a block diagonal matrix with*

$$\mathsf{B}_{\mathsf{S}_1} = \begin{pmatrix} 1/\sqrt{2} & 1 & 0 \\ 1/\sqrt{2} & 0 & 1 \end{pmatrix}. \tag{2.30}$$

The pseudo inverse $\mathsf{A}^\dagger = \mathsf{A}^T$ *is also a block diagonal matrix with* $\mathsf{B}_{\mathsf{A}^\dagger} = \mathsf{B}_{\mathsf{A}}^T$. *Since there is no overlapping between blocks, we can consider the transform for a single block with 2 inputs and 3 outputs. The eigenvalues of* $\mathsf{B}_{\mathsf{A}^\dagger}\mathsf{B}_{\mathsf{A}^\dagger}^T$ *are* $\{1, 1\}$, *while the eigenvalues of* $\mathsf{B}_{\mathsf{S}_1}\mathsf{B}_{\mathsf{S}_1}^T$ *are* $\{1, 2\}$. *Consequently,*

$$\begin{aligned} &\|\mathsf{B}_{\mathsf{A}^\dagger}\| = 1, \qquad tr(\mathsf{B}_{\mathsf{A}^\dagger}\mathsf{B}_{\mathsf{A}^\dagger}^T) = 2, \quad \textit{and} \\ &\|\mathsf{B}_{\mathsf{S}_1}\| = \sqrt{2}, \quad tr(\mathsf{B}_{\mathsf{S}_1}\mathsf{B}_{\mathsf{S}_1}^T) = 3. \end{aligned}$$

Therefore, in this case if we use the pseudo inverse instead of the usual inverse, then we would reduce the upper bound of the reconstruction square error in (2.18) by half, and the reconstruction MSE in (2.20) by two-third.

2.3.2 Eigenvalues of the LP Operators

The improvements of the pseudo inverse over the usual inverse in the last example can be generalized for all LP's with orthogonal filters. The key is to study the eigenvalues of certain matrices. We will need the following basic result on the eigenvalues.

Theorem 2.2 ([80] (p. 53)) *Suppose* A *and* B *are real matrices of size* $m \times n$ *and* $n \times m$ *respectively, with* $m \le n$. *Then* BA *has the same eigenvalues as* AB, *counting multiplicity, together with additional* $n - m$ *eigenvalues equal to* 0.

For a square matrix A, denote the set of all its eigenvalues by $\sigma(\mathsf{M})$, where an eigenvalue λ with multiplicity n is written as $\lambda^{(n)}$. These eigenvalues are the roots of the characteristic polynomial of A which is defined as $P_{\mathsf{A}}(\alpha) \triangleq \det(\alpha\mathsf{I} - \mathsf{A})$. For a polynomial matrix $\mathsf{A}(z)$, its characteristic polynomial would generally have coefficients as polynomials of z as well. Theorem 2.2 says that $P_{\mathsf{BA}}(\alpha) = \alpha^{n-m} P_{\mathsf{AB}}(\alpha)$. It can be verified that this result also holds when A and B are polynomial matrices.

Proposition 2.2 *Suppose the LP uses orthogonal filters. Then:*

(a) $\sigma(\mathsf{A}^*(z)\mathsf{A}(z)) = \{1^{(|\mathsf{M}|)}\}$, *and*

(b) $\sigma(\mathsf{S}_1(z)\mathsf{S}_1^*(z)) = \{1^{(|\mathsf{M}|-1)}, 2\}$.

Proof: Since $\mathsf{A}(z)$ represents a tight frame with frame bounds equal to 1, we have $\mathsf{A}^*(z)\mathsf{A}(z) = \mathsf{I}$. Therefore (a) follows directly. Next, consider $\mathsf{S}_1(z)\mathsf{S}_1^*(z) = \mathsf{G}(z)\mathsf{G}^*(z) + \mathsf{I}$. Since $\mathsf{G}^*(z)\mathsf{G}(z) = 1$ by the orthogonal condition, from Theorem 2.2 it follows that $\sigma(\mathsf{G}(z)\mathsf{G}^*(z)) = \{0^{(|\mathsf{M}|-1)}, 1\}$, or

$P_{\mathsf{G}(z)\mathsf{G}^*(z)}(\alpha) = (\alpha-1)\alpha^{|\mathsf{M}|-1}$. Thus,

$$\begin{aligned} P_{\mathsf{G}(z)\mathsf{G}^*(z)+\mathsf{I}}(\alpha) &= \det(\alpha\mathsf{I} - \mathsf{G}(z)\mathsf{G}^*(z) - \mathsf{I}) \\ &= P_{\mathsf{G}(z)\mathsf{G}^*(z)}(\alpha-1) \\ &= (\alpha-2)(\alpha-1)^{|\mathsf{M}|-1}, \end{aligned}$$

which implies (b). □

Recall that with orthogonal filters, the pseudo inverse of the LP is $\mathsf{A}^\dagger(z) = \mathsf{A}^*(z)$. Thus, from (2.16) and (2.21) we immediately obtain the following results.

Corollary 2.2 *Suppose the LP uses orthogonal filters. Then:*

(a) $\|\mathsf{S}_1\| = \sqrt{2}$, *and*

(b) $\|\mathsf{A}^\dagger\| = 1$.

As a result, when the noise energy is bounded, the upper bound of the reconstruction square error in (2.18) using the pseudo inverse is equal to half of the one that uses the usual inverse.

Corollary 2.3 *Suppose the LP uses orthogonal filters and its coefficients are contaminated by an additive white noise with variance* σ^2. *Then* for one level LP *the reconstruction mean square error using the usual inverse is* $MSE_1 = \sigma^2(1+1/|\mathsf{M}|)$, *while using the pseudo inverse it is* $MSE_2 = \sigma^2$. *When* the LP is iterated with J-levels, *then for the usual inverse we have*

$$MSE_1^{(J)} = \sigma^2\left(1 + \frac{1}{|\mathsf{M}|} + \ldots + \frac{1}{|\mathsf{M}|^J}\right) \to \sigma^2\frac{|\mathsf{M}|}{|\mathsf{M}|-1},$$

while for the pseudo inverse we still have $MSE_2^{(J)} = \sigma^2$.

Therefore with white noise on the LP coefficients, the reduction in the reconstruction MSE by using the pseudo inverse instead of using the usual inverse is from $(1+1/|\mathsf{M}|)$ times for one level LP up to $(1+1/(|\mathsf{M}|-1))$ times for multiple levels LP's. In particular, for the commonly used LP in 2-D with $\mathsf{M} = 2\cdot\mathsf{I}_2$, the pseudo inverse improves the signal-to-noise ratio (SNR) of the reconstruction signal from 0.97 dB (with one level LP) to 1.25 dB (with iterated LP).

Remark 2.2 *Let us clarify the difference in the performance measurements between Corollary 2.2 and Corollary 2.3. The results in Corollary 2.2 use the* maximal *eigenvalues from Proposition 2.2, whereas the results in Corollary 2.3 use the* average *of these eigenvalues. Thus, the gain factor by the pseudo inverse in Corollary 2.2 is fixed, while the gain factor in Corollary 2.3 gets smaller as* $|\mathsf{M}|$ *becomes larger.*

Finally, we have the following properties on the operators that reconstruct from the coarse and difference signals using the pseudo inverse.

Proposition 2.3 *Consider* $\mathsf{G}(z)$ *and* $\mathsf{D}(z) = \mathsf{I} - \mathsf{G}(z)\mathsf{H}(z)$. *Suppose the LP uses orthogonal filters. Then:*

(a) $\sigma(\mathsf{G}(z)\mathsf{G}^*(z)) = \{0^{(|\mathsf{M}|-1)}, 1\}$,

(c) $\mathsf{D}^*(z) = \mathsf{D}(z)$,

(b) $\mathsf{D}(z)\mathsf{D}(z) = \mathsf{D}(z)$,

(d) $\sigma(\mathsf{D}(z)) = \{0, 1^{(|\mathsf{M}|-1)}\}$.

Proof: Part (a) was proved in the proof of Proposition 2.2. Part (b) and (c) are easily verified using the orthogonal conditions: $\mathsf{G}^*(z)\mathsf{G}(z) = 1$ and $\mathsf{H}(z) = \mathsf{G}^*(z)$. Using characteristic polynomials similarly to the proof of Proposition 2.2, part (d) is proved by observing that $P_{\mathsf{D}(z)}(\alpha) = \pm P_{\mathsf{G}(z)\mathsf{G}^*(z)}(1-\alpha)$. □

As a result, the operator for the difference signal in the LP, $\boldsymbol{d} = (\mathsf{I} - \mathsf{GH})\boldsymbol{x}$, is an orthogonal projection to a subspace, which has dimension equal to $(|\mathsf{M}| - 1)/|\mathsf{M}|$ times the dimension of the signal space.[3] Such a view can also be inferred from the geometrical proof of Theorem 2.1.

2.4 Reconstruction using Projection

2.4.1 The New Reconstruction Algorithm

In this section we consider a more general case when H and G are arbitrary filters. Even though any frame operator has a pseudo inverse, for complexity reason, we will consider only the the inverses that can be realized by a fast algorithm. Motivated by the tight frame case, we focus on the reconstruction that has a structure shown in Figure 2.3. We then turn the problem around by asking for which filters such an algorithm is indeed an inverse or pseudo inverse. This has the same flavor as the filter design problem for perfect reconstruction filter banks, thus we can resort to many existing results. The following result provides an answer to our question.

Theorem 2.3

1. *The reconstruction shown in Figure 2.3 is an inverse transform of the LP* if and only if *two filters H and G are biorthogonal with respect to the sampling lattice* M*, which means the prediction operator of the LP (*GH*) is a projector, or* $\mathsf{HG} = \mathsf{I}$.

2. *Furthermore, that reconstruction is the pseudo inverse* if and only if *the prediction operator of the LP (*GH*) is an orthogonal projector.*[4]

[3] For infinite length signals, this has to be interpreted in the polyphase-domain.

[4] Recall that given a Hilbert space H, a linear operator P mapping H onto itself is called a *projector* if $P^2 = P$. Furthermore, if P is self-adjoint or $P = P^T$ then P is called an *orthogonal projector*.

Proof: 1. The transform matrix for the reconstruction algorithm in Figure 2.3 can be written as

$$\mathsf{S}_2 = \begin{pmatrix} \mathsf{G} & \mathsf{I} - \mathsf{GH} \end{pmatrix} \tag{2.31}$$

From the expression for A in (2.4), we have $\mathsf{S}_2\mathsf{A} = \mathsf{I} - \mathsf{GH} + (\mathsf{GH})^2$. Therefore S_2 is a left inverse of A if and only if $\mathsf{GH} = (\mathsf{GH})^2$ or GH is a projector.

We note that H and G possess right and left inverses, respectively (which involves inverse filters of H and G). Thus the projection condition $\mathsf{GH} = \mathsf{GHGH}$ is equivalent to

$$\mathsf{HG} = \mathsf{I} \quad \text{or} \quad \langle \tilde{h}[\cdot - \mathsf{M}k], g[\cdot - \mathsf{M}l] \rangle = \delta[k-l] \quad \forall k, l \in \mathbb{Z}^d. \tag{2.32}$$

Filters H and G satisfying (2.32) are said to be biorthogonal filters (with respect to the sampling matrix M). This proves part 1.

2. For part 2, we require additionally that S_2 is a pseudo inverse of A. From (2.13) this means that $\mathsf{A}^T\mathsf{A}\mathsf{S}_2 = \mathsf{A}^T$. Using the assumption that $\mathsf{HG} = \mathsf{I}$, after some manipulations we have that

$$\mathsf{A}^T\mathsf{A}\mathsf{S}_2 = \begin{pmatrix} \mathsf{H}^T & (\mathsf{I} - \mathsf{GH})(\mathsf{I} - \mathsf{H}^T\mathsf{G}^T) \end{pmatrix}$$

So the pseudo inverse condition is simplified to

$$(\mathsf{I} - \mathsf{GH})(\mathsf{I} - \mathsf{H}^T\mathsf{G}^T) = (\mathsf{I} - \mathsf{GH})^T \tag{2.33}$$

Notice that the left hand side of (2.33) is a symmetric matrix, thus it is the case for $(\mathsf{I} - \mathsf{GH})^T$ and GH as well. Therefore GH is an orthogonal projector, which proves the part 2. □

Remark 2.3 *It is interesting to note that the two conditions for the LP in the above proposition, i.e. projection and orthogonal projection, are exactly the same as the conditions for the improved LP's that are studied in [166]. Those conditions lead to LP with interpolation and least squares LP, respectively. The motivation for those modifications there is to minimize the prediction error* $\boldsymbol{d}$ *of the LP, whereas our motivation is to have a better reconstruction algorithm for the LP. Still, the results from [166] motivate the use of filters with the aforementioned properties for the LP.*

Remark 2.4 *The orthogonal projection case obviously includes the LP with orthogonal filters studied in the last section. It is shown in [166] that under the orthogonal projection condition, if one of the LP filter is given, then the other filter is uniquely determined.*

Therefore, the minimum requirement for the FB shown in Figure 2.3 to be a bona fide inverse of the LP is the biorthogonality condition (2.32) on the filters H and G, which can be expressed equivalently in the polyphase-domain as $\mathsf{H}(z)\mathsf{G}(z) = 1$. There exist many designs for such filters due to their role in

wavelet constructions [31, 174]. Among them, a popular choice for images is the "9-7" filters from [31, 174]. Another interesting example is based on the original LP filter suggested in [18], which is especially popular for applications in vision, together with its dual filter [5]. Tables 2.1 and 2.2 list the filter coefficients. Note that these filters are symmetric and very close to being orthogonal. As a result, the proposed reconstruction in Figure 2.3 is very close to the pseudo inverse.

n	0	±1	±2	±3	±4
$h[n]$	0.852699	0.377403	-0.110624	-0.023849	0.037828
$g[n]$	0.788486	0.418092	-0.040689	-0.064539	

Table 2.1: Coefficients for the "9-7" biorthogonal filters.

With biorthogonal filters, the LP has an interesting geometrical interpretation. Let us define two subspaces V and $\tilde{V}$ that are spanned by $\{g[\cdot - \mathsf{M}k]\}_{k\in\mathbb{Z}^d}$ and $\{\tilde{h}[\cdot - \mathsf{M}k]\}_{k\in\mathbb{Z}^d}$, respectively. These are also the column and row spaces of G and H. For all $\boldsymbol{x}$ in $l_2(\mathbb{Z}^d)$, the prediction operator in the LP, $\boldsymbol{p} = \mathsf{GH}\boldsymbol{x}$, computes a projection of $\boldsymbol{x}$ onto V. Since $\mathsf{HG} = \mathsf{I}$, the difference signal $\boldsymbol{d} = \boldsymbol{x} - \boldsymbol{p}$ has the property that

$$\mathsf{H}\boldsymbol{d} = \mathsf{H}(\boldsymbol{x} - \mathsf{GH}\boldsymbol{x}) = \mathsf{H}\boldsymbol{x} - \mathsf{H}\boldsymbol{x} = \boldsymbol{0}.$$

Therefore, $\boldsymbol{d}$ is perpendicular to $\tilde{V}$. This fact is illustrated in Figure 2.4(a). The prediction operator in the LP (GH) can be called an *oblique projector* [168] and denoted by P_V.

Let us define W the orthogonal complementary subspace of $\tilde{V}$. Then it is easy to verify that $\boldsymbol{d} = \boldsymbol{x} - P_V\boldsymbol{x}$ is a projection of $\boldsymbol{x}$ onto W such that the error is parallel to V (again, refer to Figure 2.4(a)). Denote this projection as P_W, $P_W = \mathsf{I} - \mathsf{GH}$. It is easy to see that P_W is an orthogonal projection when P_V is. In addition,

$$\begin{aligned} &(\boldsymbol{x} - P_V\boldsymbol{x}) \in W \quad \forall \boldsymbol{x}, \quad \text{and} \\ &\quad P_V\boldsymbol{x} = \boldsymbol{0} \quad \Leftrightarrow \quad \boldsymbol{x} \in W. \end{aligned} \tag{2.34}$$

Next let us compare the usual reconstruction method as in Figure 2.1(b), denoted REC-1,

$$\hat{\boldsymbol{x}}_1 = \mathsf{S}_1\boldsymbol{y} = \mathsf{G}\boldsymbol{c} + \boldsymbol{d}, \tag{2.35}$$

n	0	±1	±2	±3
$h[n]/\sqrt{2}$	0.6	0.25	-0.05	0
$g[n]/\sqrt{2}$	0.607143	0.260714	-0.053571	-0.010714

Table 2.2: Coefficients for the original LP filter and its dual filter.

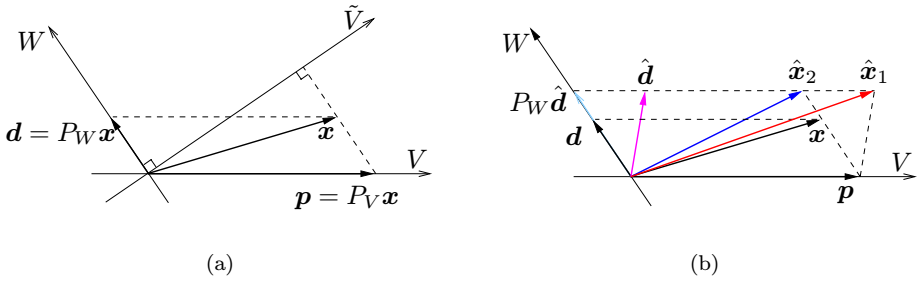


Figure 2.4: Geometrical interpretation of the LP with biorthogonal filters. (a) The LP as an oblique projector: $\boldsymbol{p} = P_V \boldsymbol{x}$ is the projection of x onto V such that the difference signal $\boldsymbol{d} = \boldsymbol{x} - P_V \boldsymbol{x}$ is perpendicular to $\tilde{V}$. (b) Comparing two reconstruction methods when $\boldsymbol{d}$ is corrupted by noise and becomes $\hat{\boldsymbol{d}}$. The usual reconstruction REC-1 adds $\hat{\boldsymbol{d}}$ directly to the reconstructed signal, while the new reconstruction REC-2 adds the P_W projection of $\hat{\boldsymbol{d}}$ and thus eliminates the error component in $\hat{\boldsymbol{d}}$ that is parallel to V.

with the new reconstruction using the FB as in Figure 2.3, denoted REC-2,

$$\hat{\boldsymbol{x}}_2 = \mathsf{S}_2 \boldsymbol{y} = \mathsf{G}\boldsymbol{c} + (\mathsf{I} - \mathsf{GH})\boldsymbol{d}. \tag{2.36}$$

These two reconstruction algorithms are different in the way of handling the difference signal $\boldsymbol{d}$. More specifically, the REC-1 method adds $\boldsymbol{d}$ directly while the REC-2 method adds the P_W projection of $\boldsymbol{d}$ to the reconstructed signal. In the absence of noise, the two methods yield identical results (as one expects) since from (2.34) we have $\boldsymbol{d} \in W$. However, as shown in Figure 2.4(b), when there is noise in the LP coefficients, the REC-2 method eliminates the influence of the error component in $\hat{\boldsymbol{d}}$ that is parallel to V.

For more quantitative measurements on the performance of the two methods, suppose that we wish to approximate $\boldsymbol{x}$ given $\hat{\boldsymbol{y}} = \mathsf{A}\boldsymbol{x} + \boldsymbol{\eta}$. With no further information about the error in the LP coefficients $\boldsymbol{\eta}$, it makes sense to chose $\hat{\boldsymbol{x}}$ that minimizes the residual $\|\mathsf{A}\hat{\boldsymbol{x}} - \hat{\boldsymbol{y}}\|$. As mentioned before, the optimal linear solution to this problem is the pseudo inverse of A. Using this as the measurement for the performance in reconstruction, the following result states that REC-2 always performs better than REC-1.

Proposition 2.4 *Assume that H and G are biorthogonal filters. Let $\hat{\boldsymbol{x}}_1$ and $\hat{\boldsymbol{x}}_2$ be the results of reconstruction from noisy LP coefficients $\hat{\boldsymbol{y}}$ using REC-1 and REC-2, respectively. Then we have*

$$\|\mathsf{A}\hat{\boldsymbol{x}}_1 - \hat{\boldsymbol{y}}\| \geq \|\mathsf{A}\hat{\boldsymbol{x}}_2 - \hat{\boldsymbol{y}}\|, \tag{2.37}$$

where equality holds if and only if $\mathsf{H}\hat{\boldsymbol{d}} = \boldsymbol{0}$.

Proof: Using the definition of A, S_1, S_2 in (2.4), (2.5), (2.31) and the fact that $\mathsf{GH} = \mathsf{I}$, we have after some manipulations

$$\begin{aligned} \mathsf{A}\hat{\boldsymbol{x}}_1 - \hat{\boldsymbol{y}} &= \mathsf{AS}_1\hat{\boldsymbol{y}} - \hat{\boldsymbol{y}} = \begin{pmatrix} \mathsf{H}\hat{\boldsymbol{d}} \\ -\mathsf{GH}\hat{\boldsymbol{d}} \end{pmatrix}, \quad \text{and} \\ \mathsf{A}\hat{\boldsymbol{x}}_2 - \hat{\boldsymbol{y}} &= \mathsf{AS}_2\hat{\boldsymbol{y}} - \hat{\boldsymbol{y}} = \begin{pmatrix} \mathbf{0} \\ -\mathsf{GH}\hat{\boldsymbol{d}} \end{pmatrix}. \end{aligned}$$

Therefore,

$$\|\mathsf{A}\hat{\boldsymbol{x}}_1 - \hat{\boldsymbol{y}}\|^2 = ||\mathsf{H}\hat{\boldsymbol{d}}||^2 + ||\mathsf{GH}\hat{\boldsymbol{d}}||^2 \geq ||\mathsf{GH}\hat{\boldsymbol{d}}||^2 = \|\mathsf{A}\hat{\boldsymbol{x}}_2 - \hat{\boldsymbol{y}}\|^2.$$

□

2.4.2 Reconstruction Error Analysis

Note that the above comparison does not give us exact behavior of the reconstruction error. In this section, we will study this error under some additional assumptions on the coefficient noise. Our analysis is complicated further since in coding, the LP is often used with quantization noise feedback where the coarse signal c is quantized *before* being fed back to the predictor. This case is referred to as the closed-loop mode in Figure 2.5. The open-loop mode refers to the case that has been considered so far, namely when the noise is added to the coefficients *after* the LP transform. A discussion of those two quantization modes in pyramid coding can be found in [133]. Note that with the closed-loop mode, we are no longer in the linear framework, so that optimality of the pseudo-inverse does not hold.[5]

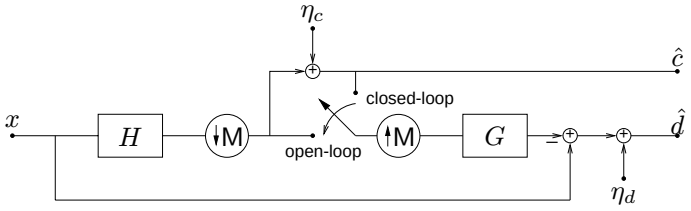


Figure 2.5: Laplacian pyramid encoding. The open-loop mode bypasses the coarse level quantization, whereas the closed-loop mode includes it.

To analyze the reconstruction error, we separate the noise in the LP coefficients into two components: $\boldsymbol{\eta} = (\boldsymbol{\eta}_c, \boldsymbol{\eta}_d)^T$, corresponding to the coarse and detail quantization as shown in Figure 2.5. In the open-loop (ol) mode, we can apply (2.17) to obtain the reconstruction errors using REC-1 and REC-2 as

$$\begin{aligned} \boldsymbol{\epsilon}_1^{(\mathrm{ol})} &= \mathsf{G}\boldsymbol{\eta}_c + \boldsymbol{\eta}_d, \quad \text{and} \\ \boldsymbol{\epsilon}_2^{(\mathrm{ol})} &= \mathsf{G}\boldsymbol{\eta}_c + (\mathsf{I} - \mathsf{GH})\boldsymbol{\eta}_d. \end{aligned} \tag{2.38}$$

[5] In that case, a *consistent reconstruction* algorithm [73] can lead to improvements.

In the closed-loop (cl) mode, the outputs of the LP after adding noise are

$$\begin{aligned}\hat{\boldsymbol{c}} &= \boldsymbol{c}+\boldsymbol{\eta}_c = \mathsf{H}\boldsymbol{x}+\boldsymbol{\eta}_c, \quad \text{and}\\ \hat{\boldsymbol{d}} &= \boldsymbol{d}+\boldsymbol{\eta}_d = \boldsymbol{x}-\mathsf{G}\hat{\boldsymbol{c}}+\boldsymbol{\eta}_d = (\mathsf{I}-\mathsf{GH})\boldsymbol{x}-\mathsf{G}\boldsymbol{\eta}_c+\boldsymbol{\eta}_d.\end{aligned}$$

Compared to the open-loop mode, an extra noisy term $-\mathsf{G}\boldsymbol{\eta}_c$ is added to the output $\hat{\boldsymbol{d}}$. From these noisy LP coefficients, the usual method REC-1 reconstructs

$$\hat{\boldsymbol{x}}_1^{(\mathrm{cl})} = \mathsf{G}\hat{\boldsymbol{c}}+\hat{\boldsymbol{d}} = \boldsymbol{x}+\boldsymbol{\eta}_d.$$

Thus the term $-\mathsf{G}\boldsymbol{\eta}_c$ is canceled in the reconstructed signal by REC-1. For the proposed method REC-2, under the biorthogonal assumption $\mathsf{GH}=\mathsf{I}$, after some manipulations we have

$$\hat{\boldsymbol{x}}_2^{(\mathrm{cl})} = \mathsf{G}\hat{\boldsymbol{c}}+(\mathsf{I}-\mathsf{GH})\hat{\boldsymbol{d}} = \boldsymbol{x}+\mathsf{G}\boldsymbol{\eta}_c+(\mathsf{I}-\mathsf{GH})\boldsymbol{\eta}_d.$$

In this case, the projector for the difference signal $(\mathsf{I}-\mathsf{GH})$ annihilates the noisy term $-\mathsf{G}\boldsymbol{\eta}_c$ in $\hat{\boldsymbol{d}}$ because $(\mathsf{I}-\mathsf{GH})\mathsf{G} = \mathsf{G}-\mathsf{GHG} = \mathsf{0}$. So the reconstruction errors using REC-1 and REC-2 in the closed-loop mode are

$$\begin{aligned}\boldsymbol{\epsilon}_1^{(\mathrm{cl})} &= \boldsymbol{\eta}_d, \quad \text{and}\\ \boldsymbol{\epsilon}_2^{(\mathrm{cl})} &= \mathsf{G}\boldsymbol{\eta}_c+(\mathsf{I}-\mathsf{GH})\boldsymbol{\eta}_d. \qquad (2.39)\end{aligned}$$

From (2.38) and (2.39) we observe that with the REC-1 method, the coarse level noise $\boldsymbol{\eta}_c$ does not effect the reconstruction signal in the closed-loop mode. This makes the usual reconstruction method attractive in coding application. However, the problem of allocating bits to the quantizers Q_c and Q_d for the coarse and difference signals becomes difficult. The reason is that in minimizing $\boldsymbol{\epsilon}_1^{(\mathrm{cl})}$ the choice for Q_d depends on the choice for Q_c, and thus one must use a complicated bit allocation technique [133]. With the REC-2 method, the reconstruction errors are the same in both open- and closed-loop modes, thus we can simply use the open-loop mode. Furthermore, in the tight frame case, because $\|\boldsymbol{\epsilon}\|^2 = \|\boldsymbol{\eta}\|^2$ the square error distortion can be minimized in the LP domain using standard bit allocation techniques such as [149].

Now suppose that $\|\boldsymbol{\eta}_c\|$ is negligible compared to $\|\boldsymbol{\eta}_d\|$. This is a reasonable assumption since $\boldsymbol{\eta}_c$ has $|\mathsf{M}|$ times fewer samples than $\boldsymbol{\eta}_d$. Furthermore, suppose that the predictor (GH) becomes an orthogonal projection which implies $\|(\mathsf{I}-\mathsf{GH})\boldsymbol{\eta}_d\| \le \|\boldsymbol{\eta}_d\|$. Under these conditions in both modes we have: $\|\boldsymbol{\epsilon}_2\| \le \|\boldsymbol{\epsilon}_1\|$, or REC-2 performs better than REC-1. A finer analysis under the additive white noise model is provided by the following result.

Proposition 2.5 *Suppose the LP use orthogonal filters and the additive noises $\boldsymbol{\eta}_c$ and $\boldsymbol{\eta}_d$ to the LP coefficients are uncorrelated white with variances σ_c^2 and σ_d^2, respectively. Then the reconstruction mean square errors by using REC-1 and REC-2 are:*

(a) For the open-loop mode:

$$\begin{aligned} MSE_1^{(ol)} &= \frac{1}{|\mathsf{M}|}\sigma_c^2 + \sigma_d^2 \\ MSE_2^{(ol)} &= \frac{1}{|\mathsf{M}|}\sigma_c^2 + \frac{|\mathsf{M}|-1}{|\mathsf{M}|}\sigma_d^2 \end{aligned}$$

(a) For the closed-loop mode:

$$\begin{aligned} MSE_1^{(cl)} &= \sigma_d^2 \\ MSE_2^{(cl)} &= \frac{1}{|\mathsf{M}|}\sigma_c^2 + \frac{|\mathsf{M}|-1}{|\mathsf{M}|}\sigma_d^2 \end{aligned}$$

Proof: Since $\boldsymbol{\eta}_c$ and $\boldsymbol{\eta}_d$ are uncorrelated and white, applying (2.21) to (2.38) we have

$$\begin{aligned} MSE_2^{(\mathrm{ol})} &= \frac{\sigma_c^2}{|\mathsf{M}|(2\pi)^d}\int_{[-\pi,\pi]^d} \mathrm{tr}(\mathsf{G}(e^{j\omega})\mathsf{G}^*(e^{j\omega}))d\omega + \\ &\quad \frac{\sigma_d^2}{|\mathsf{M}|(2\pi)^d}\int_{[-\pi,\pi]^d} \mathrm{tr}(\mathsf{D}(e^{j\omega})\mathsf{D}^*(e^{j\omega}))d\omega. \end{aligned}$$

From this, using Proposition 2.3 we get the desired result for $MSE_2^{(\mathrm{ol})}$. Other results follow similarly. □

Numerical results on real images follow the above analysis very closely, even for filters that are only approximately orthogonal such as the "9-7" biorthogonal filters in Table 2.1 (which was used in all of our experiments). For example, in the image coding application, assume that uniform scalar quantizers with equal step for coarse and difference signals, $\Delta_c = \Delta_d = \Delta$ are used, then we have $\sigma_c^2 \approx \sigma_d^2 \approx \Delta^2/12$. In this case, the new inverse REC-2 improves the SNR of the reconstructed signal over the usual inverse REC-1 by $10\log_{10}(5/4) = 0.97$ dB in the open-loop mode, while giving the same performance as the usual inverse in the closed-loop mode. Figure 2.6 shows the result for the "Barbara" image of size 512×512. However, note that the optimal choice for Δ_c and Δ_d depends on the signal.

In some other applications like denoising, the LP coefficients are thresholded so that only the M most significant coefficients are retained. (This is normally done in the open-loop mode.) In this case, it is difficult to model the coefficient noise which strongly depends on the input signal. Table 2.3 presents the numerical results for some standard test images. We observe that the new inverse consistently provides better performance by around 0.5 dB in SNR.

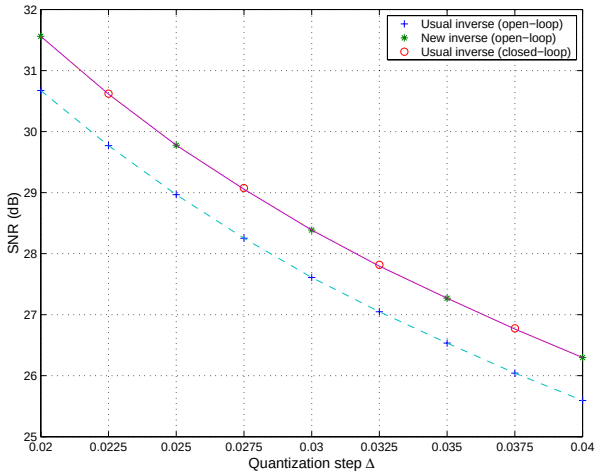


Figure 2.6: Comparison of reconstructions from the quantized LP coefficients of the "Barbara" image. The LP is decomposed with just two levels.

M		2^{12}	2^{14}	2^{16}
Barbara	REC-1	9.68	12.56	20.94
	REC-2	**9.87**	**13.18**	**21.75**
Goldhill	REC-1	12.30	15.79	21.55
	REC-2	**12.60**	**16.23**	**22.19**
Peppers	REC-1	15.06	20.81	26.77
	REC-2	**15.62**	**21.33**	**27.32**

Table 2.3: SNR's of the reconstructed signals from the M most significant LP coefficients. The image sizes are 512×512. The LP is decomposed with six levels.

2.5 Laplacian Pyramid as an Oversampled Filter Bank

2.5.1 Equivalent Filters in an Oversampled Filter Bank

The polyphase matrices for the LP operators given in Section 2.2.3 suggest that we can treat each polyphase component of the difference signal separately as being filtered and downsampled by M. We can reformulate the LP as an oversampled FB as shown in Figure 2.7. Note that every LP can be expressed as an oversampled FB, but not every oversampled filter bank in Figure 2.7 corresponds to a LP structure, since we will see that for the LP, all filters K_i are specified by H and G.

From (2.10), it follows that the polyphase vector for the equivalent filter K_i is the $(i+1)$-st row of the matrix $\mathsf{I}-\mathsf{G}(z)\mathsf{H}(z)$, which is equal to $\boldsymbol{e}_i^T - G_i(z)\mathsf{H}(z)$,

where $\boldsymbol{e}_i$ is the $(i+1)$-st column of the identity matrix. Thus,

$$K_i(z) = z^{k_i} - G_i(z^{\mathsf{M}})H(z), \quad \text{for } i = 0, \ldots, |\mathsf{M}| - 1. \tag{2.40}$$

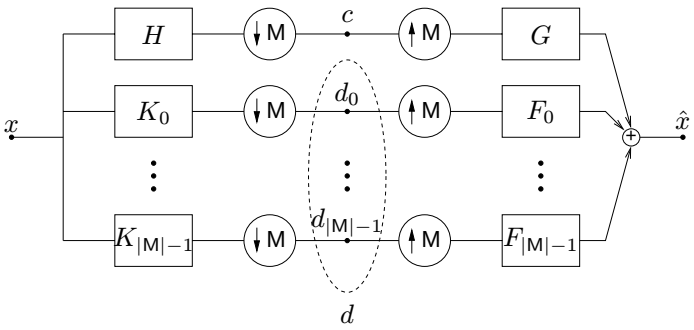


Figure 2.7: Laplacian pyramid as an oversampled filter bank, where $\{d_0, \ldots, d_{|\mathsf{M}|-1}\}$ are the polyphase components of d.

Similarly, on the synthesis side, for the REC-1 method, the equivalent filter $F_i^{[1]}$ has the polyphase vector as $\boldsymbol{e}_i$, therefore

$$F_i^{[1]}(z) = z^{-k_i}, \quad \text{for } i = 0, \ldots, |\mathsf{M}| - 1. \tag{2.41}$$

For the REC-2 method, the equivalent filters $F_i^{[2]}$ has the polyphase vector as $\boldsymbol{e}_i - \mathsf{G}(z)H_i(z)$, which implies that

$$F_i^{[2]}(z) = z^{-k_i} - G(z)H_i(z^{\mathsf{M}}), \quad \text{for } i = 0, \ldots, |\mathsf{M}| - 1. \tag{2.42}$$

Since $H(z)$ and $G(z)$ are both lowpass filters, it is easy to see that $K_i(z)$ and $F_i^{[2]}(z)$ are highpass filters, whereas $F_i^{[1]}(z)$ are allpass filters. Figure 2.8 displays the frequency responses of the equivalent filters for the LP in 1-D using the biorthogonal filter pair "9-7" from Table 2.1.

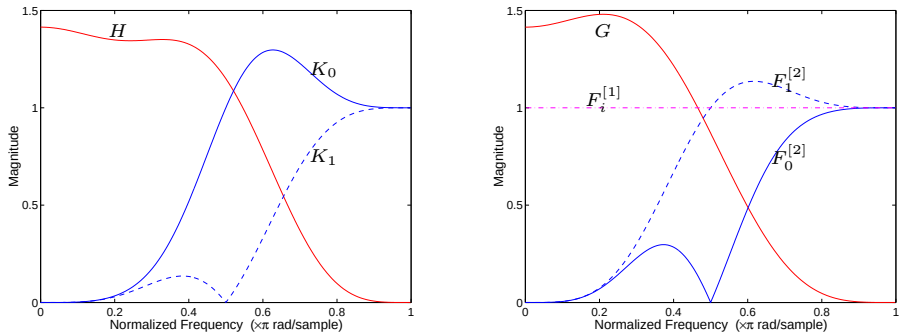
In the 1-D case with $M = 2$ and biorthogonal filters, using the property $G_0(z)H_0(z) + G_1(z)H_1(z) = 1$, we can simplify the expressions for equivalent filters in (2.40) as

$$\begin{aligned} K_0(z) &= -zH_1(z^2)G(-z) \\ K_1(z) &= +zH_0(z^2)G(-z), \end{aligned}$$

and for the synthesis filters of the REC-2 method as

$$\begin{aligned} F_0^{[2]}(z) &= -z^{-1}G_1(z^2)H(-z) \\ F_1^{[2]}(z) &= +z^{-1}G_0(z^2)H(-z) \end{aligned}$$

As a result, if the LP filters $H(e^{j\omega})$ and $G(e^{j\omega})$ are designed to have p_h and p_g zeros at $\omega = \pi$, then $K_i(e^{j\omega})$ and $F^{[2]}(e^{j\omega})$ have p_g and p_h zeros at $\omega = 0$, respectively. This result can be observed in Figure 2.8 for the "9-7" case



(a) (b)

Figure 2.8: Frequency responses of the equivalent filters for the LP using the biorthogonal filter pair "9-7". (a) Analysis. (b) Synthesis, using the usual (REC-1, denoted by superscript $^{[1]}$) and the new (REC-2, denoted by superscript $^{[2]}$) methods.

where $p_h = p_g = 4$. The number of zeros at $\omega = \pi$ of the filter determines the maximum degree of polynomials that can be reproduced by that filter and is referred to the *accuracy* number [160]. The number of zeros at $\omega = 0$ indicates the maximum degree of polynomial that are annihilated by the filters and is referred to the number of *vanishing moments*. Therefore, the LP in 1-D with high accuracy filters also has good compression properties for polynomial signals. For example, for the LP with "9-7" filters, the output d is zero whenever the input is a polynomial signal of degree up to three.

2.5.2 General Reconstruction

An interesting question is, what is the most general reconstruction for a given LP? In the polyphase-domain, this is equivalent to determine the most general form of the synthesis polyphase matrix $\mathsf{S}(z)$ such that it satisfies the following perfect reconstruction (PR) condition [169, 175]

$$\mathsf{S}(z)\mathsf{A}(z) = \mathsf{I}. \tag{2.43}$$

Corresponding to a polyphase matrix $\mathsf{S}(z)$ satisfying the PR condition (2.43) is a set of synthesis filters G and F_i for the FB in Figure 2.7 so the input signal can be perfectly reconstructed from the output of the LP. One parameterization for the left inverse of $\mathsf{A}(z)$ is given in [14] as

$$\mathsf{S}(z) = \tilde{\mathsf{S}}(z) + \mathsf{U}(z)[I - \mathsf{A}(z)\tilde{\mathsf{S}}(z)] \tag{2.44}$$

where $\tilde{\mathsf{S}}(z)$ is any particular left inverse of A, and $\mathsf{U}(z)$ is an $|\mathsf{M}| \times (|\mathsf{M}|+1)$ matrix with bounded entries. In our case, a good choice for $\tilde{\mathsf{S}}(z)$ is the usual inverse

$\mathsf{S}_1(z) = \begin{pmatrix}\mathsf{G}(z) & \mathsf{I} - \mathsf{G}(z)\mathsf{H}(z)\end{pmatrix}$. Let us split $\mathsf{S}(z)$ into two sub-matrices $\mathsf{S}_c(z)$ and $\mathsf{S}_d(z)$ of size $|\mathsf{M}| \times 1$ and $|\mathsf{M}| \times |\mathsf{M}|$, respectively: $\mathsf{S}(z) = \begin{pmatrix}\mathsf{S}_c(z) & \mathsf{S}_d(z)\end{pmatrix}$; and similarly for $\mathsf{U}(z)$: $\mathsf{U}(z) = \begin{pmatrix}\mathsf{U}_c(z) & \mathsf{U}_d(z)\end{pmatrix}$. Then the reconstruction for the LP using such $\mathsf{S}(z)$ can be written in polyphase-domain as

$$\hat{\boldsymbol{x}}(z) = \mathsf{S}_c(z)C(z) + \mathsf{S}_d(z)\boldsymbol{d}(z)$$

Substituting $\tilde{\mathsf{S}}(z) = \mathsf{S}_1(z)$ into (2.44) and after some manipulations on the block matrices, we obtain the following result.

Theorem 2.4 *The most general form for the synthesis polyphase matrix providing perfect reconstruction for a LP can be written as* $\mathsf{S}(z) = \begin{pmatrix}\mathsf{S}_c(z) & \mathsf{S}_d(z)\end{pmatrix}$ *with*

$$\begin{aligned}\mathsf{S}_c(z) &= \mathsf{G}(z) - [\mathsf{U}_c(z) - \mathsf{U}_d(z)\mathsf{G}(z)]\,[\mathsf{H}(z)\mathsf{G}(z) - 1],\\ \mathsf{S}_d(z) &= \mathsf{I} - [\mathsf{U}_c(z) - \mathsf{U}_d(z)\mathsf{G}(z)]\,\mathsf{H}(z),\end{aligned} \tag{2.45}$$

and where $\mathsf{U}_c(z)$ *and* $\mathsf{U}_d(z)$*) are matrices of size* $|\mathsf{M}| \times 1$ *and* $|\mathsf{M}| \times |\mathsf{M}|$*, respectively, with bounded entries.*

As a consequence of the above result, for a given LP, matrices $\mathsf{U}_c(z)$ and $\mathsf{U}_d(z)$ can be optimized so that the resulting synthesis filters have certain desired properties. We observe that if the LP uses biorthogonal filters satisfying $\mathsf{H}(z)\mathsf{G}(z) = 1$, then from (2.45) we have $\mathsf{S}_c(z) = \mathsf{G}(z)$. This means that all the synthesis FB's providing perfect reconstruction for the LP in this case necessarily have G as the synthesis filter for the coarse channel.

Example 2.3 *Let us consider the LP with Haar filters as in Example 2.2. In this case, we have* $H_0(z) = H_1(z) = G_0(z) = G_1(z) = 1/\sqrt{2}$*. By applying (2.45) and denoting* $V_i(z) = \left(-2\mathsf{U}_{i,1}(z) + \mathsf{U}_{i,2}(z) + \mathsf{U}_{i,3}(z)\right)/(2\sqrt{2})$*,* $i = 1, 2$*, any synthesis polyphase matrices* $\mathsf{S}(z)$ *providing PR can be written as*

$$\mathsf{S}(z) = \begin{pmatrix} G_0(z) & 1 + V_1(z) & V_1(z) \\ G_1(z) & V_2(z) & 1 + V_2(z) \end{pmatrix}.$$

Thus, the most general form of synthesis filters for the Haar LP are $G(z)$ *and*

$$\begin{aligned}F_0(z) &= 1 + V(z),\\ F_1(z) &= z^{-1} + V(z),\end{aligned}$$

with any stable filter $V(z)$*, where we denote* $V(z) = V_1(z^2) + z^{-1}V_2(z^2)$*. The usual and pseudo inverses correspond to* $V(z) = 0$ *and* $V(z) = (-1 + z^{-1})/2$*, respectively.*

2.6 Iterated Laplacian Pyramids and Derived Continuous Frames

2.6.1 Iterated Laplacian Pyramids

We now shift our focus to multilevel LP's where the scheme in Figure 2.1 is iterated on the coarse signal. The oversampled FB representation of the LP in Figure 2.7 allows us to analyze the multilevel LP as an iterated FB. Using the multirate identity, which says that filtering by $G(z)$ followed by upsampling by M is equivalent to upsampling by M followed by filtering by $G(z^{\mathsf{M}})$ [169, 175], we have the following equivalent synthesis filters at the n-level of a multilevel LP as

$$F_i^{(n)}(z) = F_i(z^{\mathsf{M}^{n-1}}) \prod_{j=0}^{n-2} G(z^{\mathsf{M}^j}), \quad i = 0, \ldots, |\mathsf{M}| - 1. \tag{2.46}$$

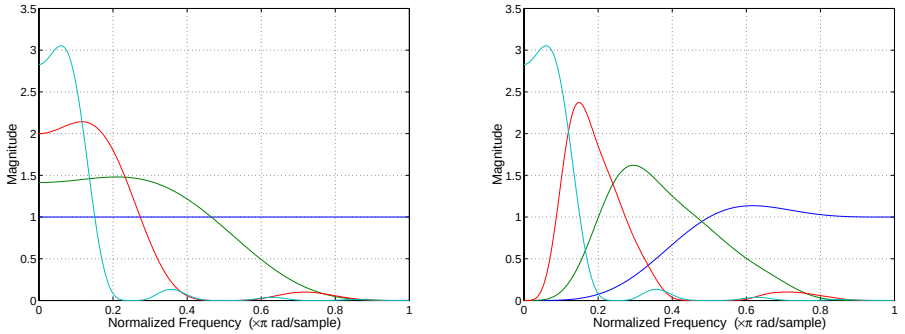
Next, consider what happens, when the synthesis filters in (2.41) and (2.42) are substituted into (2.46). Figure 2.9 shows an example of frequency responses for the equivalent filters. In the REC-1 method, we see that the synthesis functions for the LP are all low frequency signals. Thus, the errors from highpass subbands of a multilevel LP do not remain in these subbands but appear as broadband noise in the reconstructed signal. In [150], this effect was noted as the most serious disadvantage of the LP for coding applications. In the REC-2 method, the synthesis functions have similar frequency characteristics as the analysis functions, which are essentially highpass filters. Clearly, reconstruction using the REC-2 method remedies the previous mentioned problem of the LP.

The advantage of the new reconstruction method REC-2 over REC-1 is even more prominent when the errors in the LP coefficients have non-zero mean. In such case, with the REC-1 method, this non-zero mean propagates through all the lowpass synthesis filters and appears in the reconstructed signal. By contrast, with the REC-2 method, the non-zero mean is canceled by the bandpass synthesis filters. Figure 2.10 shows an example of this situation where the errors in the LP coefficients are uniformly distributed in $[0, 0.1]$.

2.6.2 Framelets from the Laplacian Pyramid

As iterated orthogonal filter banks lead to wavelets, a multilevel LP is associated with a frame for continuous functions that is called wavelet frame or framelets [139, 10]. We concentrate first on the orthogonal case. Using the multiresolution analysis framework by Mallat and Meyer [101, 112, 103], it follows that, under certain conditions, associated with the orthogonal lowpass filter G in the LP is an orthonormal scaling function $\phi(t) \in L^2(\mathbb{R}^d)$ that generates a multiresolution analysis (MRA) represented by a sequence of nested subspaces $\{V_j\}_{j\in\mathbb{Z}}$,

$$\ldots V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \ldots \tag{2.47}$$



(a) (b)

Figure 2.9: Frequency responses of the equivalent synthesis filters for the multi-level LP with "9-7" filters. (a) Usual reconstruction method REC-1. Note that all synthesis filters are lowpass. (b) New reconstruction method REC-2. Here, the synthesis filters are bandpass and match with the frequency regions of corresponding subbands, as expected. Consequently, REC-2 confines the influence of noise from the LP only in these localized bands.

with

$$\text{Closure}\left(\bigcup_{j\in\mathbb{Z}} V_j\right) = L^2(\mathbb{R}^d), \tag{2.48}$$

$$\bigcap_{j\in\mathbb{Z}} V_j = \{0\}. \tag{2.49}$$

The scaling function ϕ is specified from the filter G via the two-scale equation:

$$\phi(t) = |\mathsf{M}|^{1/2} \sum_{n\in\mathbb{Z}^d} g[n]\phi(\mathsf{M}t - n) \tag{2.50}$$

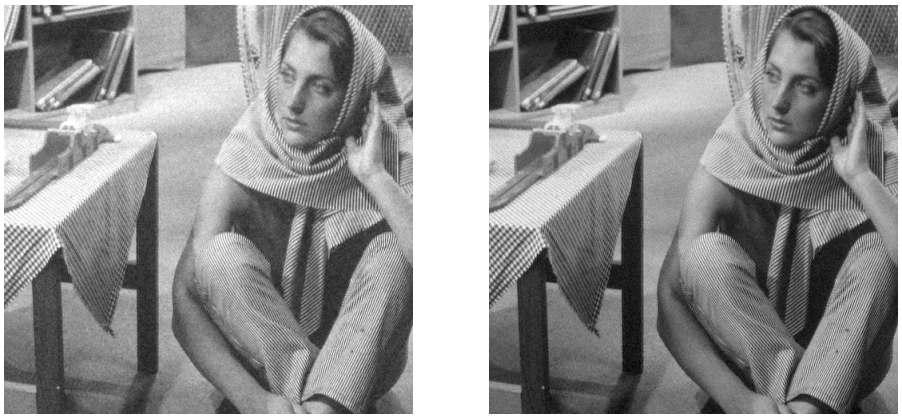
Denote

$$\phi_{j,n}(t) = |\mathsf{M}|^{-j/2}\phi(\mathsf{M}^{-j}t - n), \quad j \in \mathbb{Z}, n \in \mathbb{Z}^d. \tag{2.51}$$

Then the family $\{\phi_{j,n}\}_{n\in\mathbb{Z}^d}$ is an orthonormal basis of V_j for all $j \in \mathbb{Z}$. Define W_j to be the orthogonal complement of V_j in V_{j-1}:

$$V_{j-1} = V_j \oplus W_j. \tag{2.52}$$

Denote F_i $(0 \le i \le |\mathsf{M}| - 1)$ the equivalent synthesis filters for the new reconstruction method REC-2, which in this case are simply the time-reversed versions of the analysis filters K_i. Note that F_i are highpass filters. As in the



(a) (b)

Figure 2.10: Reconstructions from the LP coefficients that are contaminated by an additive uniform white noise in the interval $[0, 0.1]$ (the original pixel values are between 0 and 1). The LP is decomposed with 6 levels. (a) Usual reconstruction method REC-1: SNR = 6.28 dB. (b) New reconstruction method REC-2: SNR = 17.42 dB.

wavelet filter bank, we associate to each of these filters a continuous function $\psi^{(i)}$ where

$$\psi^{(i)}(t) = |\mathsf{M}|^{1/2} \sum_{n \in \mathbb{Z}^d} f_i[n] \phi(\mathsf{M}t - n). \tag{2.53}$$

These functions also generate families of scaled and translated functions as

$$\psi^{(i)}_{j,n} = |\mathsf{M}|^{-j/2} \psi^{(i)}(\mathsf{M}^{-j}t - n), \quad j \in \mathbb{Z}, n \in \mathbb{Z}^d. \tag{2.54}$$

The relationship between these functions and the computational procedure of the LP can be seen as follows. Suppose f is a function in V_{j-1}, then

$$f(t) = \sum_{n \in \mathbb{Z}^d} \underbrace{\langle f, \phi_{j-1,n} \rangle}_{c^{(j-1)}[n]} \phi_{j-1,n}(t). \tag{2.55}$$

Using the two-scale equations for ϕ and ψ, it is easy to verify that the inner products of f with functions at the next scale can be written as

$$c^{(j)}[n] = \langle f, \phi_{j,n} \rangle = \sum_{k \in \mathbb{Z}^d} c^{(j-1)}[k] g[k - \mathsf{M}n], \tag{2.56}$$

$$d^{(j)}_i[n] = \langle f, \psi^{(i)}_{j,n} \rangle = \sum_{k \in \mathbb{Z}^d} c^{(j-1)}[k] f_i[k - \mathsf{M}n]. \tag{2.57}$$

Therefore, $\{c^{(j)}[n],\ d^{(j)}[n]\}$ is exactly the output of the LP given the input sequence $c^{(j-1)}[n]$.

Theorem 2.5 *Suppose the LP with orthogonal filter generates an MRA. Then for a scale j, $\{\psi_{j,n}^{(i)}\}_{0\le i\le |\mathsf{M}|-1,\, n\in\mathbb{Z}^d}$ is a tight frame of W_j. For all scales, $\{\psi_{j,n}^{(i)}\}_{0\le i\le |\mathsf{M}|-1,\, j\in\mathbb{Z},\, n\in\mathbb{Z}^2}$ is a tight frame of $L^2(\mathbb{R}^d)$. In all cases, the frame bounds equal to 1.*

Proof: Let f be a function in W_j. This is equivalent to $f \in V_{j-1}$ and $f \perp V_j$. Therefore, f can be expanded by an orthonormal basis of V_{j-1} as in (2.55), and $c^{(j)}[n] = \langle f, \phi_{j,n}\rangle = 0$, for all $n \in \mathbb{Z}^d$.

Now suppose that f is analyzed via the LP as in (2.57). From Theorem 2.1, with orthogonal filters, the LP provides a tight frame with frame bounds equal 1, or $||c^{(j-1)}||^2 = ||c^{(j)}||^2 + \sum_{i=0}^{|\mathsf{M}|-1} ||d_i^{(j)}||^2$.

Consequently,

$$\begin{aligned} ||f||^2 &= ||c^{(j-1)}||^2 \\ &= \sum_{i=0}^{|\mathsf{M}|-1} ||d_i^{(j)}||^2 \\ &= \sum_{i=0}^{|\mathsf{M}|-1} \sum_{n\in\mathbb{Z}^d} |\langle f, \psi_{j,n}^{(i)}\rangle|^2, \end{aligned}$$

which proves the tight frame condition for W_j. The result for $L^2(\mathbb{R}^d)$ immediately follows since the MRA conditions implies that

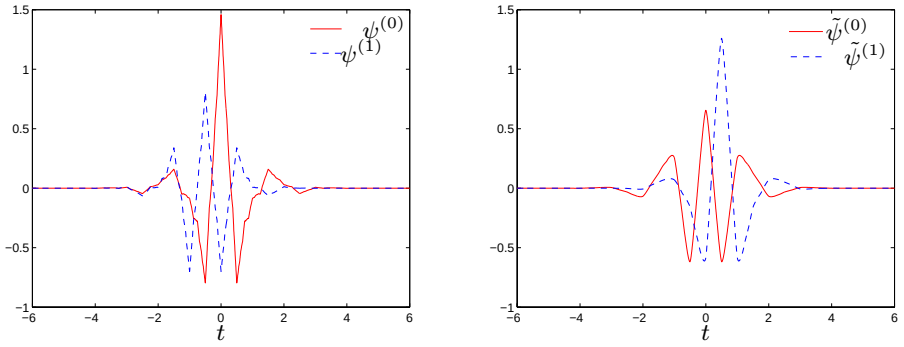
$$L^2(\mathbb{R}^d) = \bigoplus_{j\in\mathbb{Z}} W_j,$$

a decomposition of $L^2(\mathbb{R}^d)$ into mutual orthogonal subspaces. □

With this, the family $\{\psi_{j,n}^{(i)}\}_{0\le i\le |\mathsf{M}|-1,\, j\in\mathbb{Z},\, n\in\mathbb{Z}^2}$ is referred to as a tight wavelet frame or tight framelets [139]. For more general filters, with a similar setup, the LP filter bank in Figure 2.7 leads to a pair of wavelet frames – sometimes called bi-framelets [43] – generated by functions $\{\psi^{(i)}\}$ and $\{\tilde{\psi}^{(i)}\}$ for the analysis and synthesis sides, respectively. Figure 2.11 shows the 1-D bi-framelets derived from the iterated LP using "9-7" filters.

2.7 Concluding Remarks

The Laplacian pyramid was studied using the theory of frames and oversampled filter banks. We proposed a new reconstruction algorithm for the LP based on projection, which is the pseudo inverse in certain cases. The new method presents an efficient filter bank, that leads to improvements over the usual method for reconstruction in the presence of noise. With Theorem 2.1, 2.3, and 2.4, we provided a complete characterization of tight frame, reconstruction using projection, and general reconstruction for the LP, respectively. Finally, we derived the continuous-domain frames associated with the iterated LP.



(a) LP framelets

(b) Reconstruction framelets

Figure 2.11: Bi-frame from the LP with the "9-7" filters. (a) Analysis framelets. (b) Synthesis framelets using the new reconstruction method.

Our study suggests that it is useful to look at the LP from different perspectives: the standard view, the wavelet view and the oversampled filter bank view. In the wavelet view, the difference signal of the LP can be seen as the summation of the upsampled and filtered subband signals from a critical wavelet filter bank. Equivalently, the difference signal is the summation of all projections of the input signal into wavelet subspaces at a given scale. Such a view is reflected in the study of the LP with orthogonal and biorthogonal filters. In the oversampled filter bank view, the difference signal in the LP is seen a combination from multiple highpass channels in the filter bank. This view provides an effective mechanism to investigate more general structures of the LP as well as the behavior of multilevel LP's.

For practical applications, we recommend the use of the symmetric biorthogonal filters with sufficient accuracy numbers, such as the "9-7" pair for the LP, together with the new reconstruction method REC-2. The resulting LP exhibits excellent compression property for piecewise smooth signals while having a simple structure, even in higher dimensions.

Chapter 3

Iterated Directional Filter Banks

3.1 Introduction

Filter banks (FB) are powerful tools in digital signal processing. On the practical side, they provide an efficient and structured method for decomposing and analyzing discrete signals [169, 175]. On the theoretical side, filter banks play an important role that, since certain constructions can lead to multiresolution bases in the continuous domain, with wavelets as the primary example [42, 174, 160, 103].

While extensive research on FB and their related constructions for bases have been carried out in the one-dimensional (1-D) case, there are limited results in the multi-dimensional (M-D) case. On the one hand, the most common approach is to generalize 1-D systems to higher dimensions in a separable fashion. In such a case, the theory carries over straightforwardly and the implementation has low complexity. On the other hand, several studies [30, 93, 27] indicate that interesting non-separable FB and bases can be constructed in M-D. However, to our knowledge, in the past, non-separable methods have had been of limited use in applications. The reason is the high complexity and the unclear practical needs for such methods. In addition, the design issues for non-separable FB are often much more complicated.

Recently, there has been an active research effort for a new and "true" two-dimensional (2-D) image representations [173]. One of the major gaps to be filled in the existing toolkits for image processing, is to explore the intrinsic geometrical structure of typical images. Recent developments on new 2-D multiscale representations beyond wavelets are the *ridgelet* and *curvelet* constructions by Candès and Donoho [21, 20], and they indicate that directionality is a crucial feature for an efficient image representation. This result is also supported by past studies on the human visual system [44] as well as by recent studies to

[0]This chapter includes research conducted jointly with Martin Vetterli [58, 54].

identify the sparse components of natural images [117]. These results point out the directional characteristic of the response of visual cortical cells, which can be interpreted as the result of providing a sparse code for natural scenes.

In the filter bank literature, Bamberger and Smith in 1992 [9] explored non-separable FB in constructing a 2-D directional filter bank (DFB) that can be maximally decimated while achieving perfect reconstruction. However the use of the DFB so far has been limited to image analysis applications [7, 121], where the emphasis was placed on obtaining good directional selectivity filter banks.

In this work, we are interested in using the DFB as a structured linear expansion for images. In particular, we want to construct local and directional bases using the DFB. In order to do that, it is necessary to study in detail the behavior of the iterated DFB, including understanding of the equivalent filters and the sampling matrices for the DFB channels. Furthermore, we have to address the regularity issue, that ensures the convergence and smoothness of the associated bases of the iterated DFB.

The chapter begins with a brief overview of multi-dimensional multirate systems that sets up the background for later discussion. Section 3.3 discusses the DFB in detail using a new construction that is simpler and more intuitive than the original one given in [9]. This new construction allows us to give explicit formulas for the equivalent sampling matrices and filters of each channel in the iterated DFB. Section 3.4 studies the directional local bases generated by the DFB. In particular, we examine in detail the "sinc" case, where the basis functions are assumed to have ideal frequency response. This leads to a connection between the iterated DFB and a *local Radon transform* in the continuous domain. Finally, Section 3.5 concludes with some discussions and an outlook.

Notations: Vectors are denoted by bold-face symbols, while matrices are denoted by capital serif symbols. The z-transform of a multi-dimensional signal is defined as

$$X(\boldsymbol{z}) = \sum_{\boldsymbol{n} \in \mathbb{Z}^d} x[\boldsymbol{n}] \boldsymbol{z}^{-\boldsymbol{n}},$$

where raising a d-dimensional complex vector $\boldsymbol{z} = (z_0, \ldots, z_{d-1})^T$ to a d-dimensional integer vector $\boldsymbol{n} = (n_0, \ldots, n_{d-1})^T$ yields $\boldsymbol{z}^{\boldsymbol{n}} = \prod_{i=0}^{d-1} z_i^{n_i}$. The quantity z^{M} is defined to be a vector whose $(k+1)$-st element is given by z^{m_k}, where m_k is the $(k+1)$-st column of the matrix M.

The discrete-time Fourier transform of a multi-dimensional signal is defined as

$$X(\boldsymbol{\omega}) = \sum_{\boldsymbol{n} \in \mathbb{Z}^d} x[\boldsymbol{n}] e^{-j \boldsymbol{\omega}^T \boldsymbol{n}}.$$

Notice that we use the same capital letter for both the z-transform and Fourier transform of the signal.

3.2 Multi-Dimensional Multirate Systems

We start by reviewing some of the concepts and results in multi-dimensional multirate systems [172, 176, 93, 169, 175]. In particular, we examine in detail the quincunx filter bank, since it serves as the core of directional filter banks.

3.2.1 Sampling in Multiple Dimensions

In multi-dimensional multirate systems, sampling operations are defined on lattices. A lattice in d dimensions is represented by a $d \times d$ nonsingular integer matrix M as

$$LAT(\mathsf{M}) = \{\boldsymbol{m} : \boldsymbol{m} = \mathsf{M}\boldsymbol{n},\ \boldsymbol{n} \in \mathbb{Z}^d\}, \tag{3.1}$$

In other words, the indices of points belonging to the sub-lattice $LAT(\mathsf{M})$ are given as weighted integer combinations of the columns of M. The original lattice is assumed to be $\mathbb{Z}^d$. For example, the following matrices are possible representations of the two-dimensional *quincunx* sub-lattice [172] shown in Figure 3.1 in which one out of two points is retained:

$$\mathsf{Q}_0 = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}, \qquad \mathsf{Q}_1 = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \tag{3.2}$$

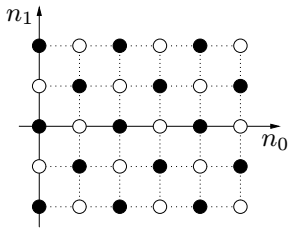


Figure 3.1: Quincunx sampling lattice.

For an M-fold *downsampling* the input $x(\boldsymbol{n})$ and the output $x_d(\boldsymbol{n})$ are related by [176, 169]

$$\begin{aligned} x_d[\boldsymbol{n}] &= x[\mathsf{M}\boldsymbol{n}] \\ X_d(\boldsymbol{\omega}) &= \frac{1}{|\det(\mathsf{M})|} \sum_{\boldsymbol{k} \in \mathcal{N}(\mathsf{M}^T)} X(\mathsf{M}^{-T}\boldsymbol{\omega} - 2\pi\mathsf{M}^{-T}\boldsymbol{k}). \end{aligned} \tag{3.3}$$

Here $\mathcal{N}(\mathsf{M})$ is defined as the set of integer vectors of the form $\mathsf{M}\boldsymbol{t}$, where $\boldsymbol{t} \in [0,1)^d$. The number of elements in $\mathcal{N}(\mathsf{M})$ is equal to $|\det(\mathsf{M})|$. The matrix $\mathsf{M}^{-T} = (\mathsf{M}^T)^{-1}$ generates the *reciprocal lattice* of $LAT(\mathsf{M})$[67], which consists of points representing the replicated spectra in the frequency domain.

For an M-fold *upsampling*, the input $x(\boldsymbol{n})$ and the output $x_u(\boldsymbol{n})$ are related by

$$\begin{aligned} x_u[\boldsymbol{n}] &= \begin{cases} x[\mathsf{M}^{-1}\boldsymbol{n}] & \text{if } \boldsymbol{n} \in LAT(\mathsf{M}), \\ 0 & \text{otherwise} \end{cases} \\ X_u(\boldsymbol{\omega}) &= X(\mathsf{M}^T\boldsymbol{\omega}), \\ X_u(\boldsymbol{z}) &= X(\boldsymbol{z}^{\mathsf{M}}). \end{aligned} \tag{3.4}$$

There are special cases when the sampling operations use unimodular integer matrices (i.e. matrices with determinant equal to ± 1). Sampling by a unimodular integer matrix does not change the data rate but only rearranges the input samples; thus it is referred to as a *resampling* operation. Furthermore it can be shown [23] that

Proposition 3.1 $LAT(\mathsf{A}) = LAT(\mathsf{B})$ *if and only if* $\mathsf{A} = \mathsf{BE}$ *where* E *is a unimodular integer matrix.*

The following four basic unimodular matrices are used in the DFB in order to provide the equivalence of the rotation operations:

$$\begin{aligned} \mathsf{R}_0 &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \qquad \mathsf{R}_1 = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}, \\ \mathsf{R}_2 &= \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \qquad \mathsf{R}_3 = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}. \end{aligned} \tag{3.5}$$

Figure 3.2 shows an example of a resampled image. Note that $\mathsf{R}_0\mathsf{R}_1 = \mathsf{R}_2\mathsf{R}_3 = \mathsf{I}_2$ (here I_2 denotes the 2×2 identity matrix) so that, for example, upsampling by R_0 is equivalent to downsampling by R_1.

(a) (b)

Figure 3.2: Example of a resampled image. (a) The "cameraman" image. (b) The "cameraman" image after being resampled by R_0.

A useful tool in analyzing multi-dimensional multirate operations is the *Smith form* that can diagonalize any integer matrix M into a product UDV where U and V are unimodular integer matrices and D is an integer diagonal

matrices [169]. The quincunx matrix in (3.2) can be expressed in the Smith form as

$$\begin{aligned} \mathsf{Q}_0 &= \mathsf{R}_1\mathsf{D}_0\mathsf{R}_2 = \mathsf{R}_2\mathsf{D}_1\mathsf{R}_1 \quad \text{and} \\ \mathsf{Q}_1 &= \mathsf{R}_0\mathsf{D}_0\mathsf{R}_3 = \mathsf{R}_3\mathsf{D}_1\mathsf{R}_0, \end{aligned} \tag{3.6}$$

where

$$\mathsf{D}_0 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathsf{D}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \tag{3.7}$$

are two 2-D diagonal matrices that correspond to dyadic sampling in each dimension.

For the interchange of filtering and sampling, multirate identities [169, 175] can be used. The identity for the analysis side of the filter bank is shown in Figure 3.3; the one for the synthesis side can be inferred similarly. Downsampling by M followed by filtering with a filter $H(\boldsymbol{\omega})$ is equivalent to filtering with the filter $H(\mathsf{M}^T\boldsymbol{\omega})$, which is obtained by upsampling $H(\boldsymbol{\omega})$ by M, before downsampling.

Figure 3.3: Multi-dimensional multirate identity for interchange of downsampling and filtering.

3.2.2 Quincunx Filter Bank

Finally, let us look at the quincunx filter bank (QFB) shown in Figure 3.4. In this chapter, we consider the quincunx sampling matrix Q to be either Q_0 or Q_1. Those sampling matrices generate the same sub-lattice but the downsampling operation rotates the input image by $-45°$ and $45°$, respectively.

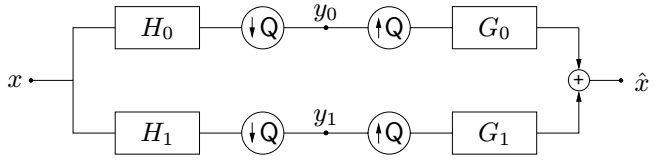


Figure 3.4: Quincunx filter bank.

Using (3.3) and (3.4), the input output relationship of the QFB can be expressed as

$$\begin{aligned} \hat{X}(\boldsymbol{\omega}) &= \frac{1}{2}\left[H_0(\boldsymbol{\omega})G_0(\boldsymbol{\omega}) + H_1(\boldsymbol{\omega})G_1(\boldsymbol{\omega})\right]X(\boldsymbol{\omega}) + \\ &\quad \frac{1}{2}\left[H_0(\boldsymbol{\omega}+\boldsymbol{\pi})G_0(\boldsymbol{\omega}) + H_1(\boldsymbol{\omega}+\boldsymbol{\pi})G_1(\boldsymbol{\omega})\right]X(\boldsymbol{\omega}+\boldsymbol{\pi}), \end{aligned} \tag{3.8}$$

where $\boldsymbol{\pi} = (\pi, \pi)^T$.

Note that there is a strong resemblance between this expression and the one for the two-channel filter bank in 1-D (see, for example, [175]). Thus, results in the 1-D counterpart can be extended to the quincunx case. In particular, the QFB provides a *biorthogonal expansion* if and only if it satisfies the following *perfect reconstruction* conditions that ensure $\hat{X}(\boldsymbol{\omega}) = X(\boldsymbol{\omega})$

$$H_0(\boldsymbol{\omega})G_0(\boldsymbol{\omega}) + H_1(\boldsymbol{\omega})G_1(\boldsymbol{\omega}) = 2, \tag{3.9}$$

$$H_0(\boldsymbol{\omega}+\boldsymbol{\pi})G_0(\boldsymbol{\omega}) + H_1(\boldsymbol{\omega}+\boldsymbol{\pi})G_1(\boldsymbol{\omega}) = 0. \tag{3.10}$$

In addition, if the synthesis filters are the reversed versions of the analysis filters then the QFB provides an *orthogonal expansion*. When the filters are restricted to be finite impulse response (FIR), the perfect reconstruction conditions imply that the synthesis filters are specified by the analysis filters (up to a shift and scale factor)

$$G_0(\boldsymbol{z}) = \boldsymbol{z}^{\boldsymbol{k}} H_1(-\boldsymbol{z}), \quad G_1(\boldsymbol{z}) = -\boldsymbol{z}^{\boldsymbol{k}} H_0(-\boldsymbol{z}). \tag{3.11}$$

The QFB can be used to split the frequency spectrum of the input signal into a lowpass and a highpass channel using a diamond-shaped filter pair, or into a horizontal and a vertical channel using a fan filter pair. Frequency characteristics of these filters are shown in Figure 3.5. Note that we can obtain one filter pair from the other by simply modulating the filters by π in either the ω_0 or ω_1 frequency variable. From (3.9) and (3.10) it can be seen that if the same modulation is applied to both analysis and synthesis filters then the perfect reconstruction conditions are preserved. Thus the design problem for the fan QFB boils down to the design of diamond-shaped QFB, and vice versa.

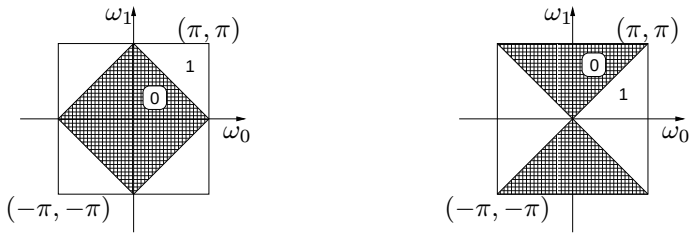


Figure 3.5: Two possible support configurations for the filters in the QFB. Each region represents the ideal frequency support of a filter in the pair. (a) Diamond-shaped filter pair. (b) Fan filter pair.

3.3 Directional Filter Banks

3.3.1 New Construction

In 1992, Bamberger and Smith [9] introduced a 2-D directional filter bank (DFB) that can be maximally decimated while achieving perfect reconstruction. The DFB is efficiently implemented via a l-level tree-structured decomposition that leads to 2^l subbands with wedge-shaped frequency partition as shown in Figure 3.6.

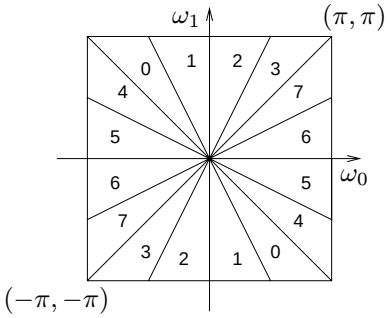


Figure 3.6: Directional filter bank frequency partitioning where $l = 3$ and there are $2^3 = 8$ real wedge-shaped frequency bands.

The original construction of the DFB in [9] involves modulating the input signal and using diamond-shaped filters. Furthermore, to obtain the desired frequency partition, an involved tree expanding rule has to be followed (see [122, 121] for details). As a result, the frequency regions for the resulting subbands do not follow a simple ordering as shown in Figure 3.6 based on the channel indices.

To simplify the analysis of the iterated DFB, in the following we propose a new formulation for the DFB that is based only on the QFB's with fan filters. The new DFB avoids the modulation of the input image and has a simpler rule for expanding the decomposition tree. We focus on the analysis side of the DFB since the synthesis is exactly symmetric. Intuitively, the wedge-shaped frequency partition of the DFB is realized by an appropriate combination of directional frequency splitting by the fan QFB's and the "rotation" operations done by resampling.

To obtain a four directional frequency partitioning, the first two decomposition levels of the DFB are given in Figure 3.7. We chose the sampling matrices in the first and second level to be Q_0 and Q_1, respectively, so that the overall sampling after two levels is $\mathsf{Q}_0\mathsf{Q}_1 = 2 \cdot \mathsf{I}_2$, or downsampling by two in each dimension.

Using the multirate identity in Figure 3.3, we can interchange the filters at the second level in Figure 3.7 with the sampling matrix Q_0. This interchange transforms a fan filter into an equivalent filter with a quadrant frequency response. Combining it with the fan filters at the first level results in the desired

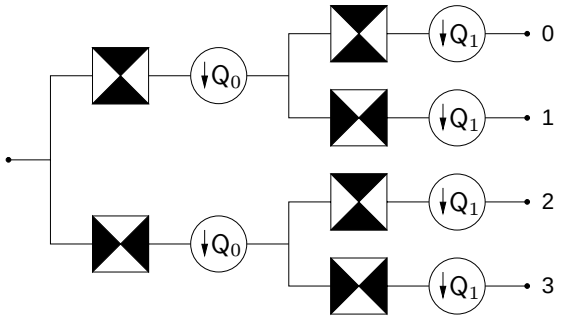


Figure 3.7: The first two levels of the DFB. At each level, QFB's with fan filters are used. The black regions represent the ideal frequency supports of the filters.

four directional subbands as shown in Figure 3.8. For later reference, we denote the equivalent filters corresponding to these four directional subbands as $K_i,\ i = 0, 1, 2, 3$.

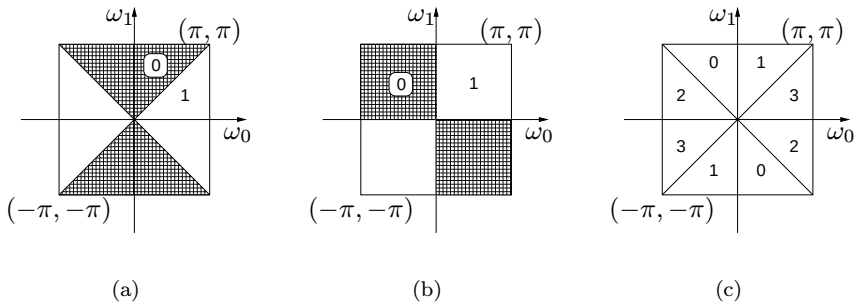


Figure 3.8: The support configuration of the equivalent filters in the first two levels of the DFB. (a) First level: fan filters. (b) Second level: quadrant filters. (c) Combining the supports of the two levels. The equivalent filters corresponding to those four directions are denoted by $K_i,\ i = 0, 1, 2, 3$.

From the third level, to achieve finer frequency partition, we use quincunx filter banks together with resampling operations as shown in Figure 3.9. There are four types of resampled QFB's, corresponding to the four resampling matrices in (3.5). Resampled QFB's of type 0 and 1 are used in the first half of DFB channels, which leads to subbands corresponding to basically horizontal directions or directions between $+45°$ and $-45°$ (for example, the frequency regions indexed by 0,1,2, and 3 in Figure 3.6). Type 2 and 3 are used in the second half of the DFB channels and lead to subbands corresponding to the remaining directions. Starting from the third level, we construct the second half of the DFB channels by simply swapping the two dimension n_0 and n_1 from the corresponding channels in the first half. This applies to both the sampling matrices (for example R_0 becomes R_2, Q_0 becomes Q_1) and the filters in the

QFB's. As a result, we only have to concentrate on the first half of the DFB channels.

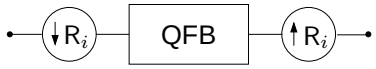
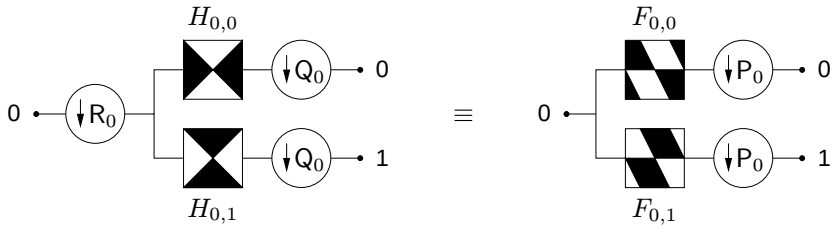
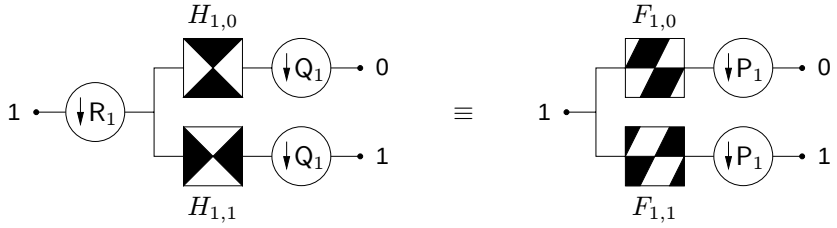


Figure 3.9: Quincunx filter banks with resampling operations that are used in the DFB starting from the third level.

The analysis side of the resampled QFB's of type 0 and 1 is detailed on the left of Figure 3.10. Notice the ordering of frequency supports of the fan filters in each QFB. The expansion rule from the third level in the first half of the channels for the iterated DFB is as follow: *at each node, the upper channel is expanded using the type 0 filter bank while the lower channel is expanded using the type 1 filter bank.*



(a) Type 0



(b) Type 1

Figure 3.10: *Left:* The analysis side of the two resampled QFB's that are used from the third level in the first half channels of the DFB. *Right:* The equivalent filter banks using parallelogram filters. The black regions represent the ideal frequency supports of the filters.

Using the multirate identity in Figure 3.3, the analysis side of the resampled QFB's can be expressed equivalently by the filter banks on the right of Figure 3.10 as being filtered by

$$F_{i,j}(\boldsymbol{\omega}) = H_{i,j}(\mathsf{R}_i^T \boldsymbol{\omega}) \tag{3.12}$$

followed by downsampling by $\mathsf{P}_i = \mathsf{R}_i\mathsf{Q}_i$, for $i, j \in \{0, 1\}$. The filters $F_{i,j}(\boldsymbol{\omega})$ obtained by resampling the fan filters $H_{i,j}(\boldsymbol{\omega})$ in (3.12) are called parallelogram filters [9].

With the Smith form of the quincunx matrices in (3.6), we can simplify the overall sampling matrices for the resampled QFB's in Figure 3.10 as

$$\begin{aligned} \mathsf{P}_0 &= \mathsf{R}_0\mathsf{Q}_0 = \mathsf{D}_0\mathsf{R}_2, \\ \mathsf{P}_1 &= \mathsf{R}_1\mathsf{Q}_1 = \mathsf{D}_0\mathsf{R}_3. \end{aligned} \tag{3.13}$$

From this by applying Proposition 3.1, we have

$$LAT(\mathsf{P}_0) = LAT(\mathsf{P}_1) = LAT(\mathsf{D}_0),$$

which means that the sampling lattices for the resampled QFB's of type 0 and 1 are equivalent to downsampling by 2 along the n_0 dimension. (Remember that we are only considering directions which are basically horizontal.)

We index the channels of a DFB with l-levels from top to bottom with the integers from 0 to $2^l - 1$, in that order. Associate with a channel in the first half indexed by k ($0 \le k < 2^{l-1}$) is a sequence of path types (where a type is either 0 or 1, see Figure 3.10) $(t_2, t_3, \ldots, t_l)$ of the filter banks from the second level leading to that channel. According to the expanding rule, $(t_2-1, t_3-1, \ldots, t_l-1)$ is the binary representation of k, or

$$k = \sum_{i=2}^{l} t_i 2^{l-i}. \tag{3.14}$$

With this path type, using the equivalent filter banks in Figure 3.10, the sequence of filtering and downsampling for the channel k can be written as

$$\boxed{K_{t_2}} - (\downarrow 2 \cdot \mathsf{I}_2) - \boxed{F_{t_2,t_3}} - (\downarrow \mathsf{P}_{t_2}) - \ldots - \boxed{F_{t_{l-1},t_l}} - (\downarrow \mathsf{P}_{t_{l-1}}). \tag{3.15}$$

From this, using the multirate identity recursively, we can transform the analysis side of the channel k ($0 \le k < 2^{l-1}$) of an l-level DFB into a single filtering with the equivalent filter $H_k^{(l)}(\boldsymbol{\omega})$ followed by downsampling by the overall sampling matrix $\mathsf{M}_k^{(l)}$, where

$$\mathsf{M}_k^{(l)} = 2 \cdot \prod_{i=2}^{l-1} \mathsf{P}_{t_i}, \tag{3.16}$$

$$H_k^{(l)}(\boldsymbol{\omega}) = K_{t_2}(\boldsymbol{\omega}) \prod_{i=2}^{l-1} F_{t_i,t_{i+1}}((\mathsf{M}_k^{(i)})^T \boldsymbol{\omega}). \tag{3.17}$$

In (3.17) the matrix $\mathsf{M}_k^{(i)}$, $(i = 2, \ldots, l)$ is understood as the partial product of the overall sampling matrix in (3.17).

3.3.2 Properties of the DFB

We offer a detailed analysis of the DFB. The following result provides the explicit expressions for the overall sampling matrices in (3.17).

Proposition 3.2 *The overall sampling matrix for the channel k ($0 \le k < 2^{l-1}$) in a l-levels DFB, $l \ge 2$, is*

$$\mathsf{M}_k^{(l)} = 2 \cdot \mathsf{D}_0^{l-2} \mathsf{R}_3^{s_l(k)}, \tag{3.18}$$

where $s_l(k) = 2\lfloor k/2 \rfloor - 2^{l-2} + 1$.

Proof: Substituting (3.13) into (3.17) and noting that $\mathsf{R}_2 = \mathsf{R}_3^{-1}$, we have

$$\begin{aligned} \mathsf{M}_k^{(l)} &= 2 \cdot \prod_{i=2}^{l-1} \left(\mathsf{D}_0 \mathsf{R}_3^{2t_i - 1} \right) \\ &= 2 \cdot \mathsf{D}_0^{l-2} \mathsf{R}_3^{\sum_{i=2}^{l-1} (2t_i - 1) 2^{l-i-1}} \end{aligned} \tag{3.19}$$

The second equality is obtained by swapping the positions of D_0 and R_3 in the matrix product using the fact that

$$\mathsf{R}_3^k \mathsf{D}_0 = \begin{pmatrix} 1 & 0 \\ -k & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -2k & 1 \end{pmatrix} = \mathsf{D}_0 \mathsf{R}_3^{2k}, \quad \forall k \in \mathbb{Z}.$$

Using the relation between k and its sequence of path type $(t_2, t_3, \ldots, t_l)$ in (3.14) we can simplify the power of R_3 in (3.19) as

$$\begin{aligned} s_l(k) &= \sum_{i=2}^{l-1} (2t_i - 1) 2^{l-i-1} \\ &= \sum_{i=2}^{l-1} t_i 2^{l-i} - \sum_{i=2}^{l-1} 2^{l-i-1} \\ &= 2\lfloor k/2 \rfloor - 2^{l-2} + 1. \end{aligned}$$

□

Remark 3.1 *Let $k' = \lfloor k/2 \rfloor$, then we can write $s_l(2k') = s_l(2k'+1) = 2k' - 2^{l-2} + 1$, which agrees with the fact that the channels $2k'$ and $2k'+1$ share the same sampling matrix since they attach to the two channels of a QFB in the last level. Because $0 \le k' < 2^{l-2}$, $s_l(k)$ is odd and $|s_l(k)| < 2^{l-2}$.*

Using the above result together with Proposition 3.1, we obtain an important property of the DFB in which the sampling lattice of its subbands (when $l \ge 2$) are diagonal or separable.

Corollary 3.1 *The overall sampling lattices of a DFB with l-levels, $l \ge 2$, is equal to $LAT(2 \cdot \mathsf{D}_0^{l-2})$ for the first half channels and $LAT(2 \cdot \mathsf{D}_1^{l-2})$ for the second half channels.*

We now ready to demonstrate that the DFB constructed as in Section 3.3.1 indeed splits the 2-D spectrum into subbands with wedge-shaped frequency partitions as shown in Figure 3.6. Denote these wedge-shaped frequency regions

by $R_k^{(l)}$, $(0 \le k < 2^l)$. Specifically, for $0 \le k < 2^{l-1}$, $R_k^{(l)}$ is the region inside the basic spectrum square $[-pi, +\pi]^2$ that is sandwiched between two lines $\omega_0 = (k - 2^{l-2})\omega_1/2^{l-2}$ and $\omega_0 = (k + 1 - 2^{l-2})\omega_1/2^{l-2}$; whereas $R_{k+2^{l-1}}^{(l)}$ is the region that is sandwiched between two lines $\omega_1 = (k - 2^{l-2})\omega_0/2^{l-2}$ and $\omega_1 = (k + 1 - 2^{l-2})\omega_0/2^{l-2}$.

Theorem 3.1 *Suppose that a tree structured DFB with l-levels uses QFB's with ideal fan filters. Then the equivalent filter $H_k^{(l)}(\boldsymbol{\omega})$ for the channel k, $0 \le k < 2^l$, has ideal frequency response in the region $R_k^{(l)}$. This mean, the first half of the channels of the DFB scan the basically horizontal directions between 45° and -45°, while the second half of the channels scan the basically vertical directions between 45° and 135°.*

Proof: We will prove this by induction. The cases $l = 1$ and $l = 2$ were already shown in Section 3.3.1 for the first two decomposition levels of the DFB. Suppose that the equivalent filters $H_k^{(l)}(\boldsymbol{\omega})$, $0 \le k < 2^{l-1}$ for a l-levels DFB has the desired frequency responses. The $(l+1)$-levels DFB is obtained by appending to each channel of a l-levels DFB a QFB according to the expanding rule. We consider two cases where k is even and odd.

(i) $k = 2k'$**:** Then a type 0 resampled QFB is appended to the channel k in the level l. This leads to two channels in the level $l + 1$, indexed by $4k'$ and $4k' + 1$, with the following equivalent filters:

$$\begin{aligned} H_{4k'}^{(l+1)}(\boldsymbol{\omega}) &= H_{2k'}^{(l)}(\boldsymbol{\omega})\, F_{0,0}((\mathsf{M}_{2k'}^{(l)})^T\boldsymbol{\omega}) \\ H_{4k'+1}^{(l+1)}(\boldsymbol{\omega}) &= H_{2k'}^{(l)}(\boldsymbol{\omega})\, F_{0,1}((\mathsf{M}_{2k'}^{(l)})^T\boldsymbol{\omega}) \end{aligned} \tag{3.20}$$

(ii) $k = 2k' + 1$**:** Then a type 1 resampled QFB is appended to the channel k in the level l. This leads to two channels in the level $l + 1$, indexed by $4k' + 2$ and $4k' + 3$, with the following equivalent filters:

$$\begin{aligned} H_{4k'+2}^{(l+1)}(\boldsymbol{\omega}) &= H_{2k'+1}^{(l)}(\boldsymbol{\omega})\, F_{1,0}((\mathsf{M}_{2k'+1}^{(l)})^T\boldsymbol{\omega}) \\ H_{4k'+3}^{(l+1)}(\boldsymbol{\omega}) &= H_{2k'+1}^{(l)}(\boldsymbol{\omega})\, F_{1,1}((\mathsf{M}_{2k'+1}^{(l)})^T\boldsymbol{\omega}) \end{aligned} \tag{3.21}$$

Using the expressions for $\mathsf{M}_k^{(l)}$ in (3.18) it can be verified that in (3.20) and (3.21), upsampling by $\mathsf{M}_k^{(l)}$ effectively squeezes and shears the basic spectrum of the parallelogram filters $F_{i,j}(\boldsymbol{\omega})$ (which are given in Figure 3.10) so that the copy that overlaps with the frequency support of $H_k^{(l)}(\boldsymbol{\omega})$ is illustrated in Figure 3.11. Clearly, this divides the frequency support of $H_k^{(l)}(\boldsymbol{\omega})$ into finer slices. Thus, the equivalent filters $H_k^{(l+1)}(\boldsymbol{\omega})$ at the next level also has the desired frequency responses.

□

In practice, filters with non-ideal frequency responses are used. Nevertheless, the upsampling operation by $\mathsf{M}_k^{(l)}$ applied on the filters $F_{i,j}$ shears their impulse

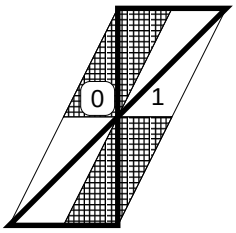


Figure 3.11: Illustration on how the frequency is divided into finer regions by the next decomposition level in the DFB. The region inside the thick lines represents the ideal frequency support of the equivalent filter after l levels. The regions inside the thin lines are copies of the basic spectrum of the upsampled parallelogram filters.

responses to different directions, which in turn produces equivalent filters that have linear supports in space and span all directions. Figure 3.12 demonstrates this fact by showing the impulse responses of 32 equivalent filters for the first half channels of a 6-levels DFB that uses Haar filters.

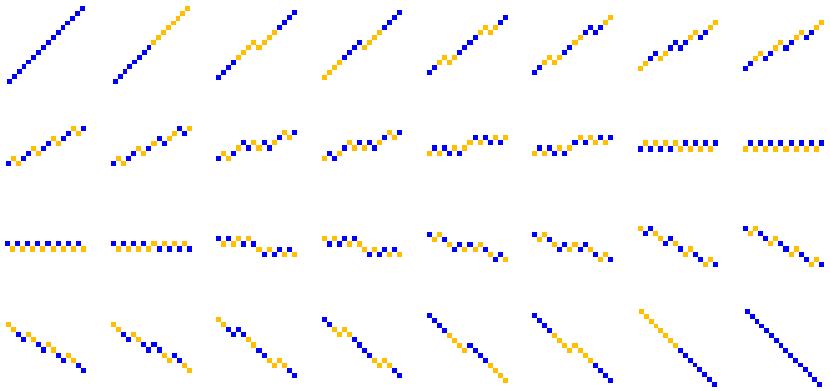


Figure 3.12: Impulse responses of 32 equivalent filters for the first half channels of a 6-levels DFB that use the Haar filters. Black and gray squares correspond to $+1$ and -1, respectively.

The original DFB [9] was known to generate visually distorted subband images, as the modulation and resampling operations introduce "frequency scrambling". In [8], the modulation problem is remedied by using the equivalent and modulated filters at each level of the DFB. For the resampling problem, Park et al. [122] propose the use of *backsampling* operations at the end of the analysis side of the DFB so that the overall sampling matrices for all channels are diagonal. Their back sampling matrices have to be computed recursively along the branches of the expanding DFB tree.

Our new formulation for the DFB already solves the modulation problem. To correct the resampling problem, the expressions for the overall sampling matrices in (3.18) leads to an explicit formula for the backsampling matrices in

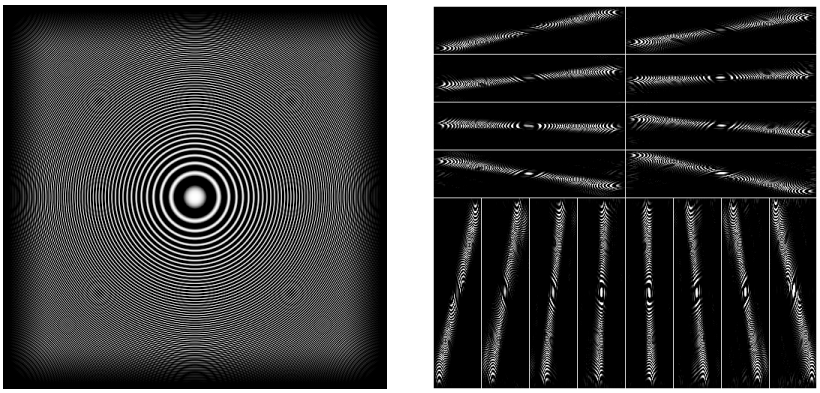
our construction as follows

$$\mathsf{B}_k^{(l)} = \mathsf{R}_3^{-s_l(k)} = \begin{pmatrix} 1 & 0 \\ s_l(k) & 1 \end{pmatrix}. \tag{3.22}$$

Again, these matrix are for the first half channels with $0 \le k < 2^{l-1}$; the ones for the second half channels are obtained by transposing these matrices. By appending a downsampling by $\mathsf{B}_k^{(l)}$ at the end of the analysis side of the channel k in the DFB, it becomes equivalent to filtering by $H_k^{(l)}(\boldsymbol{z})$ followed by downsampling by $\mathsf{S}_k^{(l)} = \mathsf{M}_k^{(l)}\mathsf{B}_k^{(l)}$, where

$$\mathsf{S}_k^{(l)} = \begin{cases} \mathrm{diag}(2^{l-1}, 2) & \text{for } 0 \le k < 2^{l-1} \\ \mathrm{diag}(2, 2^{l-1}) & \text{for } 2^{l-1} \le k < 2^l \end{cases} \tag{3.23}$$

Since $\mathsf{B}_k^{(l)}$ are unimodular matrices, sampling by these matrices only rearranges the coefficients in the DFB subbands, which enhances its visualization. Figure 3.13 shows an example of the DFB. The test image is the synthesis "zone plate" image $\cos(r^2)$ [86] where the directional frequencies match with the spatial locations. The subband images from the DFB clearly show the correct directional frequency partitioning that was mentioned.



(a) (b)

Figure 3.13: Example of the DFB. (a) The "zone plate" image. (b) The "zone plate" image decomposed by a DFB with 4 levels that leads to 16 subbands. The QFB's use the biorthogonal FIR quincunx filters designed by Phoong et al. [129] with support sizes equal to 23 × 23 and 45 × 45.

3.3.3 Filter Design

The following proposition relates the conditions for the DFB to be biorthogonal or orthogonal with respect to its kernel QFB's.

Proposition 3.3 *The DFB provides a biorthogonal or orthogonal expansion* if and only if *its kernel QFB's are biorthogonal or orthogonal, respectively.*

Proof: Since the resampling operations in the DFB are invertible, the DFB is a perfect reconstruction filter bank if and only if the QFB's used at each level are perfect reconstruction. This proves the biorthogonal condition.

The orthogonality additionally requires the synthesis filters to be the time-reversed version of the analysis filters. By matching the analysis equivalent filters of the DFB in (3.17) with the synthesis equivalent filters (which are symmetrical) it is easy to see that the DFB is orthogonal if and only if its QFB's is. □

As a result, the design issue for the DFB essentially amounts to the design of QFB's with the desired properties. In particular, we only have to design one QFB with diamond-shaped filters and modulate the filters to obtain the QFB's with fan filters for the use in the DFB.

For synthesizing biorthogonal quincunx systems, there are two main approaches [99]. In the first approach [3], the McClellan transformation [109] is used to map 1-D biorthogonal two channel filter banks to 2-D biorthogonal solutions. The second approach is based on the polyphase mapping and leads to efficient filter banks that uses the ladder structure where all the filtering operations are separable [129]. For orthogonal systems, one can use the cascade structures [93] or design directly in the time domain, possible with the help of Gröbner bases [120, 97].

3.3.4 Adaptive DFB's

The DFB considered so far employs a full tree decomposition for a particular number of levels. Finer directional selectivity requires larger number of decomposition levels, which in turn implies sparser sampling lattices. Consequently, there is a trade-off between the directional resolution and the partial resolution in the DFB. Thus for some applications, we propose the use of adaptive DFB's, in which the decisions of splitting each node on the filter bank tree depend on the input signals. This is similar to the *wavelet packets* approach [35, 134]. The potential of this scheme that is it can offers a rich menu of discrete bases, from which the "best" one can be chosen ("best" according to a particular criterion). Moreover, as with wavelet packets, the best basis can be found via a fast dynamic programming algorithm.

3.4 Directional Local Bases

3.4.1 Discrete Series Expansion from the DFB

The results from Section 3.3.2 prompt us to consider a multi-channel directional filter bank (MDFB) (see Figure 3.14) which results from transforming an

l-levels DFB into 2^l channels with equivalent filters and diagonal sampling matrices given in (3.23). Obviously, all the channels in the MDFB have the same sampling density, which is equal to the number of channels, as $\det(\mathsf{S}_k) = 2^l$, for $k = 0, \ldots, 2^l - 1$. The multi-channel view allows us to study in depth the structure of the basis generated by a DFB. Unless stated otherwise, in this section we consider the number levels l of the DFB to be fixed, $l \geq 2$, and thus the superscript $^{(l)}$ can be omitted.

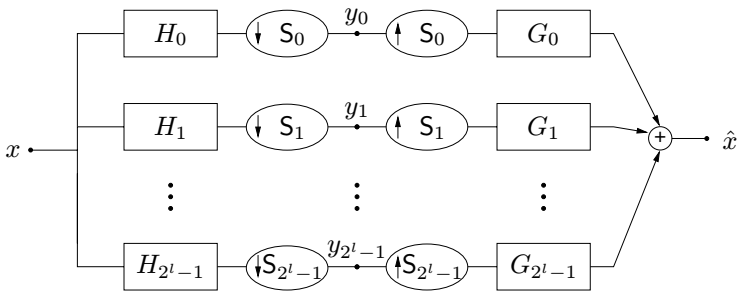


Figure 3.14: The multi-channel view of a l-levels DFB that has 2^l channels with equivalent filters and diagonal sampling matrices.

We recall that the frequency region associated to the filters $H_k(\boldsymbol{\omega})$ and $G_k(\boldsymbol{\omega})$ in Figure 3.14 is R_k that was defined just before Theorem 3.1. A fundamental property of the MDFB is that shifting the frequency region R_k on the reciprocal lattice of S_k tiles the frequency plane, as illustrated in Figure 3.15. More specially, denoting δ_R the characteristic function of the region R, i.e. $\delta_R(\boldsymbol{\omega})$ equals to 1 if $\boldsymbol{\omega} \in R$ and 0 otherwise, then it is easy to verify that

$$\sum_{\boldsymbol{m} \in \mathbb{Z}^2} \delta_{R_k}(\boldsymbol{\omega} - 2\pi \mathsf{S}_k^{-T} \boldsymbol{m}) = 1 \quad \text{for all} \quad k = 0, \ldots, 2^l - 1;\ \boldsymbol{\omega} \in \mathbb{R}^2. \tag{3.24}$$

Suppose that the filter bank in Figure 3.14 is perfect reconstruction, then an arbitrary signal $x[\boldsymbol{n}]$ in $l^2(\mathbb{Z}^2)$ can be written as

$$x[\boldsymbol{n}] = \sum_{k=0}^{2^l-1} \sum_{\boldsymbol{m} \in \mathbb{Z}^2} y_k[\boldsymbol{m}] g_k[\boldsymbol{n} - \mathsf{S}_k \boldsymbol{m}], \tag{3.25}$$

where

$$y_k[\boldsymbol{m}] = \langle x, h_k[\mathsf{S}_k \boldsymbol{m} - \cdot] \rangle. \tag{3.26}$$

Moreover, the expansion in (3.25) is unique. Therefore, the family

$$\{g_k[\boldsymbol{n} - \mathsf{S}_k \boldsymbol{m}]\}_{0 \leq k < 2^l,\, \boldsymbol{m} \in \mathbb{Z}^2} \tag{3.27}$$

which is obtained by translating the impulse responses of the synthesis filters G_k over the sampling lattices S_k, is a *basis* for discrete signals in $l^2(\mathbb{Z}^2)$. The family $\{h_k[\mathsf{S}_k \boldsymbol{m} - \boldsymbol{n}]\}_{0 \leq k < 2^l,\, \boldsymbol{m} \in \mathbb{Z}^2}$ obtained similarly from the time-reversed version

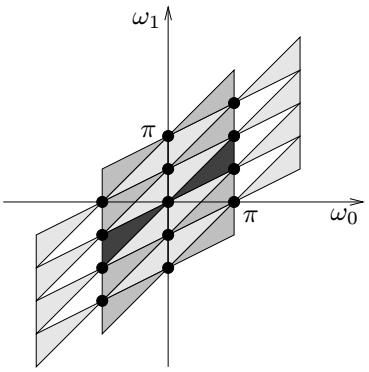


Figure 3.15: Illustration of the tiling property of the frequency regions for the filters in the multi-channel DFB. Shifting the ideal frequency region (in black) on the reciprocal lattice (represented by the black points) would tile the frequency plane.

of the analysis filters is called the *dual basis*. By substituting $x[\boldsymbol{n}] = g_k[\boldsymbol{n} - \mathsf{S}_k\boldsymbol{m}]$ into (3.25) and invoking the uniqueness of the expansion we obtain the following *biorthogonal* relation between these two bases:

$$\langle g_k[\cdot - \mathsf{S}_k\boldsymbol{m}], h_{k'}[\mathsf{S}_{k'}\boldsymbol{m}' - \cdot]\rangle = \delta[k - k']\delta[\boldsymbol{m} - \boldsymbol{m}']. \tag{3.28}$$

The expansion (3.25) becomes *orthogonal* when $g_k[\boldsymbol{n} - \mathsf{S}_k\boldsymbol{m}] = h_k[\mathsf{S}_k\boldsymbol{m} - \boldsymbol{n}]$, which means the synthesis filters are time-reversed version of the analysis filters. The name *directional local basis* is justified for the family $\{g_k[\boldsymbol{n} - \mathsf{S}_k\boldsymbol{m}]\}_{0\le k<2^l,\, \boldsymbol{m}\in\mathbb{Z}^2}$ since the filter G_k exhibits directional property and the signal $g_k[\boldsymbol{n} - \mathsf{S}_k]$ is locally supported around $\mathsf{S}_k\boldsymbol{m}$. In this basis, k and $\boldsymbol{m}$ are the direction and location indices, respectively.

3.4.2 Two Extreme Cases: Polyphase and Sinc Expansions

We will illustrate in this section how the series expansion is realized in the MDFB with two representative cases: one uses the "delay" filters and the other one uses "sinc" filters. The first case also demonstrates how the MDFB can be decomposed into polyphase components, which is not so obvious since the filter bank uses a mixture of two sampling lattices. The second case shows how the spectrum is split by the MDFB in the ideal case.

The key idea for the polyphase decomposition of the MDFB is to assign the two sampling matrices S_k, $0 \le k < 2^{l-1}$, and S_k, $2^{l-1} \le k < 2^l$, to the two separate cosets of the quincunx lattice. An example is given in Figure 3.16 where $l = 3$ and thus there are 8 polyphase components. Formally, we define the representative point for the polyphase component k as

$$\boldsymbol{p}_k = \begin{cases} (k,k)^T & \text{if } 0 \le k < 2^{l-1}, \\ (k+1-2^{l-1}, k-2^{l-1})^T & \text{if } 2^{l-1} \le k < 2^l. \end{cases} \tag{3.29}$$

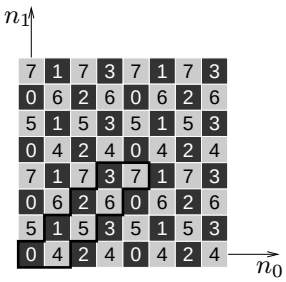


Figure 3.16: Example of polyphase decomposition in the MDFB where $l = 3$ and thus there are 8 polyphase components. Points indexed by k $(0 \leq k < 8)$ form the k-th polyphase component and they live in a sub-lattice represented by $\mathsf{S}_k^{(3)}$. Notice that points of same color live in a quincunx sub-lattice.

Lemma 3.1 *For each $\boldsymbol{n} \in \mathbb{Z}^2$, there exists a unique set pair $(k, \boldsymbol{m})$ where $k \in \{0, \ldots, 2^l - 1\}$, $\boldsymbol{m} \in \mathbb{Z}^2$ such that*

$$\boldsymbol{n} = \boldsymbol{p}_k + \mathsf{S}_k \boldsymbol{m} \tag{3.30}$$

Proof: We will prove this by uniquely constructing $(k, \boldsymbol{m})$ given $\boldsymbol{n}$. If $0 \leq k < 2^{l-1}$, equation (3.30) becomes

$$\begin{cases} n_0 = k + 2^{l-1} m_0, \\ n_1 = k + 2m_1. \end{cases}$$

This requires $n_0 + n_1$ to be even. In this case, k and m_0 are determined as the remainder and the quotient of dividing n_0 by 2^{l-1}, respectively; while $m_1 = (n_1 - k)/2$, which is an integer since $n_0 + n_1$ is even and $n_0 = k \pmod 2$. Similarly, if $2^{l-1} \leq k < 2^l$, equation (3.30) becomes

$$\begin{cases} n_0 = k + 1 - 2^{l-1} + 2m_0, \\ n_1 = k - 2^{l-1} + 2^{l-1} m_1. \end{cases}$$

This requires $n_0 + n_1$ to be odd and determines $k - 2^{l-1}$ and m_1 to be the remainder and the quotient of dividing n_1 by 2^{l-1}, respectively; while $m_0 = (n_0 - k - 1 + 2^{l-1})/2$. □

As a result, when the filters G_k are simply delays $G_k(\boldsymbol{z}) = \boldsymbol{z}^{-\boldsymbol{p}_k}$, the basis in (3.27) can be written as $\{\delta[\boldsymbol{n} - \boldsymbol{p}_k - \mathsf{S}_k \boldsymbol{m}]\}_{0 \leq k < 2^l,\, \boldsymbol{m} \in \mathbb{Z}^2}$, which is exactly the same as the standard basis for $l^2(\mathbb{Z}^2)$. The MDFB with such filters implements a polyphase transform that decompose a signal $x[\boldsymbol{n}]$ into 2^l sub-signals $\{x_k[\boldsymbol{n}]\}_{k=0,\ldots,2^l-1}$, where

$$x_k[\boldsymbol{n}] = x[\boldsymbol{p}_k + \mathsf{S}_k \boldsymbol{n}]. \tag{3.31}$$

We now turn to the case where the filters in the MDFB have ideal frequency response. Theorem 3.1 shows that such filters are obtained by iterating the DFB where QFB's with ideal fan filters are used at each level. We refer to this as the "sinc" case for the DFB. Consider

$$G_k(\boldsymbol{\omega}) = 2^{l/2} \cdot e^{j\langle \boldsymbol{\omega}, \boldsymbol{p}_k \rangle} \cdot \delta_{R_k}(\boldsymbol{\omega}). \tag{3.32}$$

The first term for G_k is a scale factor so that $\|g_k[\boldsymbol{n}]\| = 1$. The second term corresponds to a shift so that the impulse response $g_k[\boldsymbol{n}]$ is centered at $\boldsymbol{p}_k$. This shift is important for the completeness of the basis in $l^2(\mathbb{Z}^2)$. Taking the inverse discrete-time Fourier transform of (3.32) yields

$$g_k[\boldsymbol{n} - \boldsymbol{p}_k] = \frac{2^{l/2}}{(2\pi)^2} \int_{R_k} e^{j\boldsymbol{\omega}^T \boldsymbol{n}} d\boldsymbol{\omega}$$

For $0 \leq k < 2^{l-1}$, the region R_k can be specified such that the range of ω_0 depends linearly on ω_1. With this, after some manipulations we obtain an explicit expression for the impulse response of the ideal MDFB filters as

$$\begin{aligned} g_k[\boldsymbol{n} - \boldsymbol{p}_k] &= \frac{2^{l/2}}{(2\pi)^2} \int_0^{\pi} d\omega_1 \int_{(k-2^{l-2})\omega_1/2^{l-2}}^{(k+1-2^{l-2})\omega_1/2^{l-2}} d\omega_0 \left(e^{j\boldsymbol{\omega}^T \boldsymbol{n}} + e^{-j\boldsymbol{\omega}^T \boldsymbol{n}} \right) \\ &= \frac{2^{l/2}}{2\pi n_0} \left[\psi\left(\left(\frac{k+1}{2^{l-2}} - 1 \right) n_0 + n_1 \right) \right. \\ &\qquad \left. - \psi\left(\left(\frac{k}{2^{l-2}} - 1 \right) n_0 + n_1 \right) \right] \end{aligned} \tag{3.33}$$

where

$$\psi(x) = \frac{1 - cos(\pi x)}{\pi x} \tag{3.34}$$

which behaves similarly to a 1-D sinc function. Therefore, $g_k[\boldsymbol{n}]$ can be seen as the samples of the difference between two ridge functions with sinc-like ridge profiles, which are oriented around the line $n_1 = -(k/2^{l-2} - 1)n_0$, and damped by $1/n_0$. For $2^{l-1} \leq k < 2^l$, the impulse responses are obtained similarly by swapping the two dimensions n_0 and n_1.

Proposition 3.4 *With the filters G_k given in (3.32), the family (3.27) is an orthonormal basis for signals from $l^2(\mathbb{Z}^2)$.*

Proof: This result can be obtained by applying Proposition 3.3 to the "sinc" case. However, we will prove it directly by demonstrating orthogonality of the family (3.27) as well as completeness. First, the orthogonal condition requires that

$$\langle g_k(\cdot - \mathsf{S}_k \boldsymbol{m}), g_{k'}(\cdot - \mathsf{S}_{k'} \boldsymbol{m}') \rangle = \delta[k - k'] \delta[\boldsymbol{m} - \boldsymbol{m}']. \tag{3.35}$$

Since the frequency supports of the filters G_k do not overlap, (3.35) immediately holds for different filters. When $k = k'$, the left hand side of (3.35) can be

seen as an S_k-fold downsampling of the autocorrelation of $g_k[\boldsymbol{n}]$. In the Fourier domain, the autocorrelation of $g_k[\boldsymbol{n}]$ equals to $|G_k(\boldsymbol{\omega})|^2 = 2^l \delta_{R_k}(\boldsymbol{\omega})$. Therefore using (3.3), our problem reduces to

$$\sum_{\boldsymbol{m} \in \mathcal{N}(\mathsf{S}_k^T)} \delta_{R_k}(\mathsf{S}_k^{-T}\boldsymbol{\omega} - 2\pi \mathsf{S}_k^{-T}\boldsymbol{m})) = 1,$$

which comes from (3.24) after a change of variable, and thus orthogonality is proven.

For completeness, suppose it is not true, that is, suppose that there exists a signal $x[\boldsymbol{n}]$ with $\|x\| > 0$, such that

$$\langle x, g_k[\cdot - \mathsf{S}_k \boldsymbol{m}] \rangle = 0 \quad \text{for all } k, \boldsymbol{m}. \tag{3.36}$$

For a fixed k, the left side of (3.36) can be interpreted as $x[\boldsymbol{n}]$ convolved with $g_k[-\boldsymbol{n}]$, followed by a downsampling by S_k. So using (3.3) we can write (3.36) equivalently in the Fourier domain as

$$\sum_{\boldsymbol{m} \in \mathcal{N}(\mathsf{S}_k^T)} X(\boldsymbol{\omega} - 2\pi \mathsf{S}_k^{-T}\boldsymbol{m})\, G_k(-\boldsymbol{\omega} - 2\pi \mathsf{S}_k^{-T}\boldsymbol{m}) = 0, \quad \forall \boldsymbol{\omega} \in \mathbb{R}^2.$$

Since the support of $G_k(\boldsymbol{\omega})$ is non-overlapping with its alias due to downsampling by S_i as is demonstrated in (3.24), the last equation implies $X(\boldsymbol{\omega}) = 0$, $\forall \boldsymbol{\omega} \in R_k$. Combining these we have $X(\boldsymbol{\omega}) = 0$, $\forall \boldsymbol{\omega} \in [-\pi, \pi]^2$ or $\|x\| = 0$. This contradicts the assumption that $\|x\| > 0$, and thus the set is complete. □

In summary, in this section we have examined two extreme bases that can be constructed from the MDFB: one has perfect localization in time domain and the other perfect localization in the frequency domain. More interesting cases exists between these extremes using better filters.

3.4.3 Bases in the Continuous Domain

Under certain regularity conditions, the discrete bases generated by the DFB will leads to, through iteration of the number of levels l, to bases in the continuous domain. We will demonstrate this through the "sinc" case. For each filter given in (3.32) define a continuous function $\theta_k^{(l)}(\boldsymbol{t})$ by interpolating the sequence $g_k^{(l)}[\boldsymbol{n}]$ on a grid of intervals $2^{-l} \times 2^{-l}$ with a normalized 2-D sinc interpolator (so it is of norm 1)

$$\phi^{(l)}(\boldsymbol{t}) = 2^l \frac{\sin(2^l \pi t_0)}{2^l \pi t_0} \frac{\sin(2^l \pi t_1)}{2^l \pi t_1}. \tag{3.37}$$

The Fourier transform of $\theta_k^{(l)}(\boldsymbol{t})$ can be written as

$$\Theta_k^{(l)}(\boldsymbol{\omega}) = 2^{-l} \delta_{\{[-2^l\pi,\, 2^l\pi]^2\}} G_k^{(l)}[2^l \boldsymbol{\omega}].$$

Hence, for $0 \leq k < 2^{l-1}$ (the case where $2^{l-1} \leq k < 2^l$ follows similarly by swapping the two dimensions), $\Theta_k^{(l)}(\boldsymbol{\omega})$ is an indicator function for the region inside the square $[-2^l\pi,\ 2^l\pi]^2$ and is sandwiched between two lines $\omega_0 = (k - 2^{l-2})\omega_1/2^{l-2}$ and $\omega_0 = (k + 1 - 2^{l-2})\omega_1/2^{l-2}$. Similarly to (3.33), taking the inverse Fourier transform yields

$$\theta_k^{(l)}(\boldsymbol{t} - 2^{-l}\boldsymbol{p}_k^{(l)}) = \theta^{(l)}(t_0, t_1 - (2^{-(l-2)}k - 1)t_0). \tag{3.38}$$

where,

$$\theta^{(l)}(\boldsymbol{t}) = \frac{2^{l/2}}{2\pi t_0}[\psi(4t_0 + 2^l t_1) - \psi(2^l t_1)]. \tag{3.39}$$

In this case, functions corresponding to different directions (i.e. different values of k) are obtained by "shearing" a prototype function. Figure 3.17 shows a 3-D plot of the function $\theta^{(l)}(\boldsymbol{t})$. We observe a main ridge along the line $t_1 = 0$.

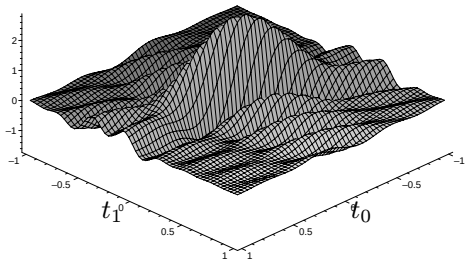


Figure 3.17: Prototype function $\theta^{(3)}(\boldsymbol{t})$ for the continuous bases resulting from a "sinc" DFB with $l = 3$. To enhance the visualization, only the region $\boldsymbol{t} \in [-1, 1]^2$ is shown. Outside this region, the function decays rapidly.

Denote $BL([-2^l\pi,\ 2^l\pi]^2)$ the space of two-dimensional real and bandlimited functions to $2^l\pi$ (that means functions whose Fourier transform vanish outside the square $[-2^l\pi,\ 2^l\pi]^2$). Using the ideal frequency characteristic of the functions $\theta_k^{(l)}(\boldsymbol{t})$, in exactly the same way as the proof for Proposition 3.4, we obtain the following result.

Proposition 3.5 *The family*

$$\left\{\theta_k^{(l)}(\boldsymbol{t} - 2^{-l}\mathsf{S}_k^{(l)}\boldsymbol{m})\right\}_{0 \leq k < 2^l,\, \boldsymbol{m} \in \mathbb{Z}^2}$$

is an orthonormal basis for $BL([-2^l\pi,\ 2^l\pi]^2)$.

Therefore a bandlimited function $f(\boldsymbol{t}) \in BL([-2^l\pi,\ 2^l\pi]^2)$ can be expressed by series expansion

$$f(\boldsymbol{t}) = \sum_{k=0}^{2^l-1} \sum_{\boldsymbol{m} \in \mathbb{Z}^2} \langle f, \theta_k^{(l)}(\cdot - 2^{-l}\mathsf{S}_k^{(l)}\boldsymbol{m})\rangle\ \theta_k^{(l)}(\boldsymbol{t} - 2^{-l}\mathsf{S}_k^{(l)}\boldsymbol{m}), \tag{3.40}$$

which can be seen as an alternative to the traditional sampling scheme by the isometric tensor-product function in (3.37).

Now consider the situation when l goes to infinity. The basis in Proposition 3.5 then becomes a basis for $\mathbb{R}^2$. The prototype function $\theta^{(l)}(\boldsymbol{t})$ can be seen by "squeezing" the function in Figure 3.17 along the axis t_1. Formally, since

$$|\psi(x)| = |\frac{1 - cos(\pi x)}{\pi x}| \leq \frac{2}{\pi |x|}, \quad \text{for } x \neq 0,$$

it follows that for $t_1 \neq 0$

$$\begin{aligned} |2^{l/2}\psi(2^l t_1)| &\leq \frac{2}{\pi 2^{l/2}|t_1|} \\ &\rightarrow 0 \quad \text{as } l \rightarrow \infty. \end{aligned}$$

Applying this to (3.39) gives

$$\lim_{l \rightarrow \infty} \theta^{(l)}(\boldsymbol{t}) = 0, \quad \text{for } t_1 \neq 0.$$

While along the line $t_1 = 0$,

$$\theta^{(l)}(t_0, 0) = 2^{l/2} \frac{\psi(4t_0)}{2\pi t_0}$$

With this, we see that the expansion (3.40) resembles a local Radon transform for function in $\mathbb{R}^2$.

3.5 Discussion

The directional filter bank studied in this chapter is a powerful mechanism for decomposing images into local and directional expansions. The filter bank is implemented efficiently with a tree structure. Especially when the quincunx filters can be realized by separable filters in the polyphase domain, the DFB has the same complexity as the 2-D wavelet packet transform. In the next chapter, the DFB is combined with a multiscale decomposition that leads to an efficient representation for piecewise smooth functions resembling images.

Since all the equivalent filters in the DFB are derived from the filters in the kernel QFB, an interesting and important question is what condition should one imposes on the filters of the QFB so that iterations in DFB converge to smooth or regular functions? In general, the regularity issue for 2-D iterated filter banks is much more involved than in the 1-D case. For directional filter banks, the problem is even more complicated since iterations are carried out in all branches. We have demonstrated the convergent of the "sinc" case to regular functions for all branches, see Figure 3.17. Figure 3.18 shows a preliminary result on the search for compactly supported regular filters for the DFB.

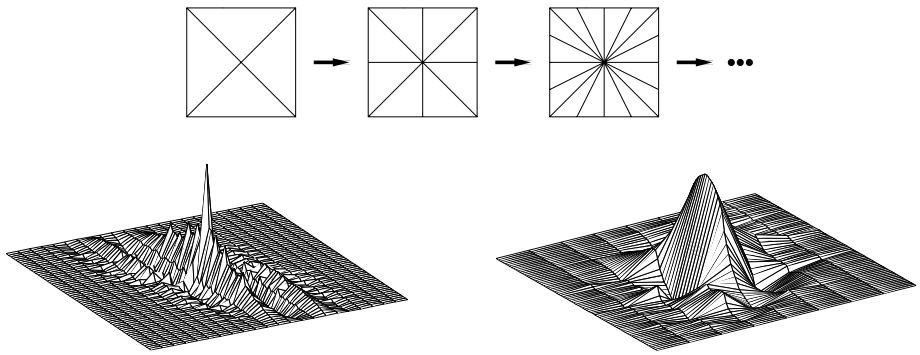


Figure 3.18: Examples of different regularity behaviors in the iterated directional filter banks. On the left is a known regular filter for the iterated quincunx filter bank [30] but fails in the DFB. On the right is a newly designed filter which is the shortest orthogonal filter that converges to a regular function in one branch of the iterated DFB.

Chapter 4

Curvelets and Filter Banks

4.1 Motivation

In this chapter, we focus on the construction of *efficient linear expansions* for *two-dimensional functions which are smooth away from discontinuities across smooth curves*. Such functions resemble natural images where discontinuities are generated by *edges* – referred to the points in the image where there is a sharp contrast in the intensity, whereas edges are often gathered along smooth *contours* which are created by (typically) smooth boundaries of physical objects. Efficiency of a linear expansion means that the coefficients for functions belonging to the class of interest are *sparse*, and thus it implies efficient representations for such functions using a non-linear approximation (NLA) scheme.

Over the last decade, wavelets have had a growing impact on signal processing, mainly due to their good NLA performance for piecewise smooth functions in one dimension (1-D). Unfortunately, this is not the case in two dimensions (2-D). In essence, wavelets are good at catching point or zero-dimensional discontinuities, but as already mentioned, two-dimensional piecewise smooth functions resembling images have one-dimensional discontinuities. Intuitively, wavelets in 2-D obtained by a tensor-product of one dimensional wavelets will be good at isolating the discontinuity at an edge point, but will not see the smoothness along the contour. This indicates that more powerful representations are needed in higher dimensions.

This fact has a direct impact on the performance of wavelets in many applications. As an example, for the image denoising problem, state-of-the-art techniques are based on thresholding of wavelet coefficients [61, 103]. While being simple, these methods work very effectively, mainly due to the property of the wavelet transform that most image information is contained in a small number of significant coefficients – around the locations of singularities or image edges. However, since wavelets fail to represent efficiently singularities along lines or curves, wavelet-based techniques fail to explore the geometrical structure that is typical in smooth edges of images. Therefore, new denoising schemes

[0]This chapter includes research conducted jointly with Martin Vetterli [58, 54].

which are based on "true" two-dimensional transforms are expected to improve the performance over the current wavelet-based methods.

Recently, Candes and Donoho [20] pioneered a new system of representation, named *curvelet*, that was shown to achieve optimal approximation behavior in a certain sense for 2-D piecewise smooth functions in $\mathbb{R}^2$ where the discontinuity curve is a C^2 function.[1] More specifically, an M-term non-linear approximation for such piecewise smooth functions using curvelets has L^2 square error decaying like $O(M^{-2})$, and this is the best rate that can be achieved by a large class of approximation processes [62]. An attractive property of the curvelet system is that such correct approximation behavior is simply obtained via thresholding a *fixed* transform.

Back to the image denoising problem, there are other approaches that explore the geometrical regularity of edges, for example by chaining adjacent wavelet coefficients and then thresholding them over these contours [104, 105]. However, the curvelet transform approach, with its "built-in" geometrical structure, provide a more direct way – by simply thresholding significant curvelet coefficients – in denoising images with smooth edges.

The original construction of the curvelet transform [20] is based on windowing the subband images into blocks and applying the ridgelet transform [19, 21] on these blocks. We will show that this approach poses several problems when one tries to implement the curvelet transform for discrete images and uses it in applications. Furthermore, as the curvelet transform was originally defined in the frequency domain, it is not clear how curvelets are sampled in the spatial domain. In fact, in [64], one of the fundamental research challenges for curvelets stated as: "is there a spatial domain scheme for refinement which, at each generation doubles the spatial resolution as well as the angular resolution?". This is what we will try to explore in the following.

One of the main subject of this chapter is that we establish a link between curvelets and filter banks and propose a new filter bank structure, named *pyramidal directional filter banks* (PDFB), that can achieve a curvelet-like image decomposition. As as a result, the PDFB provides an effective algorithmic implementation of the curvelet transform for digital images. Moreover, the proposed filter bank defines a new sampling pattern for the curvelet expansion in the spatial domain, which is entirely specified on rectangular grids. The PDFB uses appropriate combination of the Laplacian pyramid and the directional filter banks that were studied in Chapter 2 and Chapter 3, respectively.

4.2 Curvelets Background

4.2.1 Original Construction

In a nutshell, the curvelet transform [22] is obtained by filtering and then applying a windowed ridgelet transform on each bandpass image. In $\mathbb{R}^2$, ridgelets are constant along ridge lines $x_1 \cos(\theta) + x_2 \sin(\theta) = const$ and are wavelets (with

[1] C^p is the space of functions that are bounded and p-times continuously differentiable.

a scale s) along the orthogonal direction. In frequency domain, such ridgelet function is essentially localized in the corona $|\boldsymbol{\omega}| \in [2^s, 2^{s+1}]$ and around the angle θ. The ridgelet transform to provide a sparse representation for smooth objects with straight edges. A brief review of the ridgelet transform is given in Section 5.2.

To sum up, the curvelet decomposition composes of the following steps [22]:

1. Subband decomposition of the object into a sequence of subbands.
2. Windowing each subband into blocks of appropriate size, depending on its center frequency.
3. Applying the ridgelet transform on these blocks.

The motivation behind the curvelet transform is that by smooth windowing, segments of smooth curves would look straight in subimages, hence they can be well captured by a local ridgelet transform. Subband decomposition is used to keep the number of ridgelets at multiple scales under control by the fact that ridgelets of a given scale live in a certain subband. The window's size and subband frequency are coordinated such that curvelets have support obeying the key *anisotropy scaling relation for curves* [22]:

$$width \propto length^2. \tag{4.1}$$

The scaling relation (4.1) for curves is illustrated in Figure 4.1. Suppose that we want to analyze a C^2 curve at a given point. We define the coordinate system such that the v-axis is a tangent line and the u-axis is on the normal vector of the curve at the analyzed point. Then the curve can be parameterized as $u = u(v)$. Since $u(v) \in C^2$, the Taylor series expansion gives

$$u(v) \approx u(0) + u'(0)v + \frac{u''(0)}{2}v^2, \quad \text{when } v \approx 0. \tag{4.2}$$

Furthermore, because the v-axis is tangent to the curve at the point $(0, 0)$, it follows that $u(0) = u'(0) = 0$ and $u''(0) = \kappa$ which is the curvature of the curve at that point. Thus we can rewrite (4.2) as

$$u(v) \approx \frac{\kappa}{2}v^2, \quad \text{when } v \approx 0. \tag{4.3}$$

Next suppose that the underlying 2-D function that have discontinuities along the curve $u = u(v)$ is locally approximated by a basis function with rectangular support of size $l \times w$, then it follows that $w = u(l/2)$ (see Figure 4.1). Using (4.3), we obtain the following relation between the width w and the length l of the basis function

$$w \approx \frac{\kappa}{8}l^2. \tag{4.4}$$

Consequently, in a correct multiscale representation for curve discontinuities, when the scale becomes finer, the width of the basis functions should be refined at a quadric speed of the length. Curvelets provide such a representation.

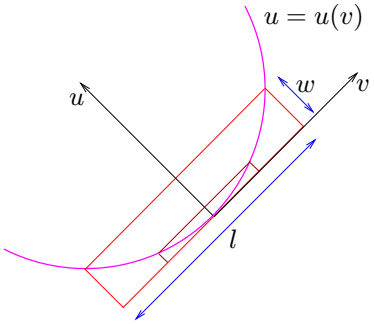


Figure 4.1: The anisotropy scaling relation for curves. The rectangular supports of the basis functions that fit a curve exhibit the quadric relation: $width \propto length^2$.

There are two important remarks at this point. First, if the discontinuity curves are only assumed to be in C^2, then approximating them with linear structures with rectangular support, shown in Figure 4.1 is the best that can be done. This is the reason why curvelets provide the optimal approximation rate for such 2-D piecewise smooth functions. However, when the discontinuity curves are known to be more regular, the curvelet representation is no longer optimal. For example with C^3 curves, an expansion with elements having parabolic shapes fitted along the curves can provide a non-linear approximation rate of $O(M^{-3})$. Secondly, the curve scaling relation (4.1) has to be interpreted locally since in (4.4) the leading constant depends on the local curvature of the discontinuity curve.

4.2.2 Non-linear Approximation Behaviors

We next sketch illustrations on the non-linear approximation behaviors of different representation systems. Rather than being rigorous, the following discussion aims at providing an intuition that can serve as a guideline for our construction of the pyramidal directional filter banks latter. For a complete and rigorous discussion, we refer to [62].

Consider a simple "Horizon" mode of piecewise smooth functions $f(x_1, x_2)$ defined on the unit square $[0, 1]^2$:

$$f(x_1, x_2) = 1_{\{x_2 \geq c(x_1)\}} \qquad 0 \leq x_1, x_2 \leq 1,$$

where the boundary of two pieces (or the contour) $c(x_1)$ is in C^p and has finite length inside the unit square. Clearly, such 2-D function has complexity equivalent to a 1-D function, namely its contour $c(x_1)$. The reason for studying this model is that the approximation rates for 2-D piecewise smooth functions resembling images are typically dominant by the discontinuity curves.

An *oracle* system, which knows precisely the contour curve $c(x_1)$, would do the best to approximate this curve when approximating the underlying 2-D function $f(x_1, x_2)$. Because f takes value either 0 or 1, the approximation errors

for surface and curve are related by

$$\begin{aligned} ||f - \hat{f}||^2 &= \int_{[0,1]^2} (f - \hat{f})^2 \\ &\leq \int_{[0,1]} |c - \hat{c}| \\ &\leq \left(\int_{[0,1]} (c - \hat{c})^2 \right)^{1/2} \\ &= ||c - \hat{c}||. \end{aligned}$$

where the second inequality comes from the Cauchy-Schwartz's inequality. Since the curve $c(x_1) \in C^p$, the optimal approximation rate is achieved by an M-term linear approximation using the Fourier basis or a wavelet basis with enough vanishing moment. These provide $||c - \hat{c}_M||^2 \sim O(M^{-2p})$ [65]. Thus the optimal approximation rate for the 2-D piecewise smooth function f is

$$||f - \hat{f}_M^{(optimal)}||^2 \sim O(M^{-p}). \tag{4.5}$$

Let us consider how a wavelet system performs for such function. Assume that the orthonormal wavelet transform with the separable Haar wavelet is employed. At the level j, wavelet basis functions have support on dyadic square of size 2^{-j} (see Figure 4.2(a)). Call n_j the number of dyadic squares at level j that intersect with the contour on the unit square. Since the contour has finite length, it follows that

$$n_j \sim O(2^j). \tag{4.6}$$

Thus, there are $O(2^j)$ nonzero wavelet coefficients at the scale 2^{-j}. This is the problem of the separable wavelet transform for 2-D piecewise smooth functions. For the 1-D piecewise smooth function, the number of significant wavelet coefficients at each scale is bounded by a constant; in the 2-D case this number grows exponentially as the scale gets finer. The total number of nonzero wavelet coefficients up to the level J is

$$N_J = \sum_{j=0}^{J} n_j \sim O(2^J). \tag{4.7}$$

Along the discontinuity curve c, it is easy to see that these nonzero wavelet coefficients decay like $O(2^{-j})$ at the j-th level. Next suppose that we keep only $M = N_J$ nonzero coefficients up to the level J in the wavelet expansion. Then the error due to truncation of the wavelet series is

$$\begin{aligned} ||f - \hat{f}_M^{(wavelet)}||^2 &\sim \sum_{j=J+1}^{\infty} 2^j (2^{-j})^2 \\ &\sim O(2^{-J}). \end{aligned} \tag{4.8}$$

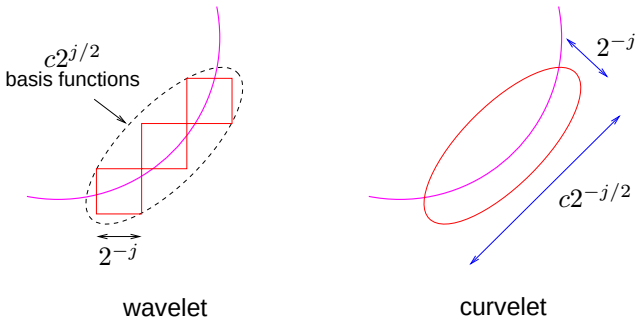


Figure 4.2: Non-linear approximation of a 2-D piecewise smooth function using wavelets and curvelets. Curvelet basis functions can be viewed as a local grouping of wavelet basis functions into linear structures so that they can capture the smooth discontinuity curve more efficiently.

Combining (4.7) and (4.8) we obtain the following non-linear approximation rate of the wavelet expansion for the "Horizon" model

$$||f - \hat{f}_M^{(wavelet)}||^2 \sim O(M^{-1}). \tag{4.9}$$

Therefore, when the discontinuity curves c is sufficiently smooth, $c \in C^p$ with $p > 1$, wavelet approximation is suboptimal. It is important to note that the smoothness of the discontinuity curve is irrelevant to the performance of the wavelet approximation. In comparison, the approximation rate for the "Horizon" functions using the Fourier basis is [63]

$$||f - f^{(Fourier)}||^2 \sim O(M^{-1/2}). \tag{4.10}$$

Therefore, it would be interesting to examine if we can improve the performance of the wavelet representation when the discontinuity curve is known to be smooth? Examining the wavelet transform in Figure 1.4 and Figure 1.6 suggests that rather than treating each significant wavelet coefficient along the discontinuity curve independently, one should group the nearby coefficients since their locations are locally correlated. Recall that at the level j, the essential support of the wavelet basis functions has size 2^{-j}. The curve scaling relation (4.1) suggests that we can group about $c2^{j/2}$ nearby wavelet basis functions into one basis function with a linear structure so that its width is proportional to its length squared (see Figure 4.2). This grouping operation reduces the number of significant coefficients at the level j from $O(2^j)$ to $O(2^{j/2})$. Consequently, this new representation provides the same approximation error as wavelets in (4.8) with only $M' \sim \sum_{j=0}^{J} 2^{j/2}$ or $O(2^{J/2})$ coefficients. In other words, the M-term non-linear approximation using this improved wavelet representation decays like

$$||f - \hat{f}_M^{(improved-wavelet)}||^2 \sim O(M^{-2}). \tag{4.11}$$

Comparing with (4.5) and (4.9), we see that for C^2 discontinuity curves, the new representation is superior compared to wavelets and actually achieves

the optimal rate. The curvelet system achieves this optimality using a similar argument. In the original curvelet construction [20], the linear structure of the basis function comes from the ridgelet basis while the curve scaling relation is ensured by suitable combination of subband filtering and windowing.

Alternatively, we could link the significant wavelet coefficients along the *entire* discontinuity curve and represent this curve as a smooth 1-D function. This is the spirit of the *bandelet* construction by Le Pennec and Mallat [125]. A similar approach is taken by Dragotti and Vetterli with their *edgeprint* data structure for image compression [66]. Clearly, such scheme resembles the previous oracle-based method, and thus provides better approximation rate when the discontinuity curve is more regular as with $c \in C^p$,

$$||f - \hat{f}_M^{(bandelet)}||^2 \sim O(M^{-p}). \tag{4.12}$$

Note that, however the bandelet expansion is an adaptive scheme and requires a detection of the discontinuity curve before hand.

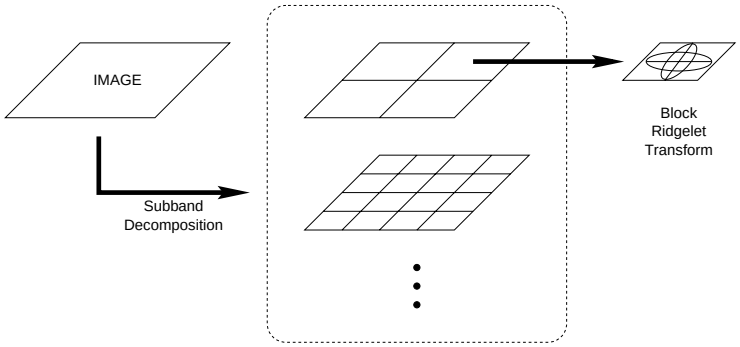
4.2.3 Curvelets and Filter Banks

The original approach for curvelet decomposition [20] poses several problems in practical applications. First, since it is a block-based transform, either the approximated images have blocking effects or one has to use overlapping windows and thus increase the redundancy. Secondly, the use of ridgelet transform, which is defined on polar coordinate, makes the implementation of the curvelet transform for discrete images on rectangular coordinates very challenging. In [64, 159, 6], different interpolation approaches were proposed to solve the polar versus rectangular coordinate transform problem, all requiring overcomplete systems. Consequently, the version of the discrete curvelet transform in [159] for example has a redundancy factor equal to $16J + 1$ where J is the number of multiscale levels. This results in the fact that curvelets are very limited in certain applications such as compression.

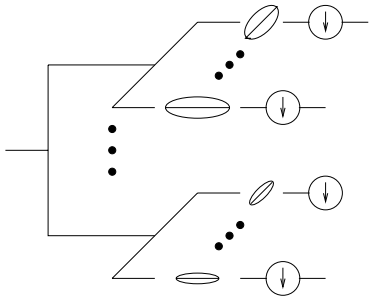
To overcome the problem of block-based approach, one could use a filter bank approach [175, 169] instead, in very much the same way as the lapped transforms [106]. The relation between the two approaches is depicted in Figure 4.3. The filter bank approach as in the lapped transform can solve the blocking effect while being critically sampled.

The grouping of wavelet coefficients argument in the last section suggests that we can have a curvelet-like representation (and thus achieve the optimal approximation rate) by first applying a multiscale decomposition and applying a local Radon transform to gather the basis functions in the same scale into linear structures. The local Radon decomposition can be obtained by a directional filter bank [9]. That is, we first use a wavelet-like decomposition for *edge (points)* detection, and then a local directional bases for *contour segments* detection.

Therefore, we can achieve a curvelet-like decomposition by a *double filter bank* in which a multiscale decomposition is used to capture the point discontinuities (i.e. edge detection), followed by a directional decomposition to link



(a)



(b)

Figure 4.3: Two approaches for curvelet decomposition. (a) *Original block-based approach:* block ridgelet transforms are applied to subband images. (b) *Proposed filter bank approach:* image is decomposed by a double filter bank structure. In this case, basis functions are given by the filter impulse responses and their translates with respect to the subsampling grids. The key issue here is to find a correct subsampling.

point discontinuities into linear structures. In this approach, the curve scaling relation is ensured by a suitable coordination between the scale and the support of the directional basis, which in turn is linked to the number of directions.

4.3 Pyramidal Directional Filter Banks

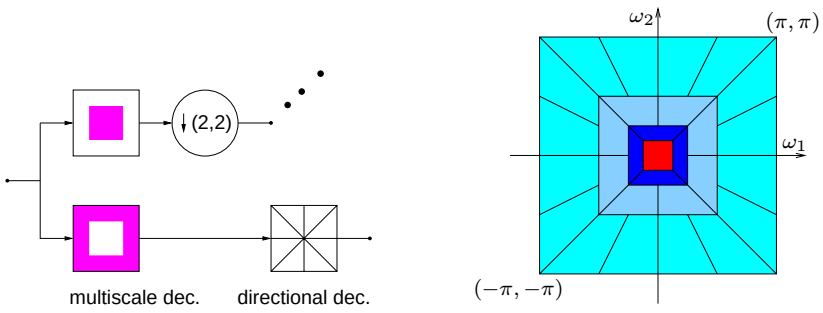
4.3.1 Multiscale and Directional Decomposition

The directional filter bank (DFB) is designed to capture the high frequency components (representing directionality) of images. Therefore, low frequency components are handled poorly by the DFB. In fact, with the frequency partition

shown in Figure 3.6, low frequencies would "leak" into several directional subbands, hence DFB does *not* provide a sparse representation for images. To improve the situation, low frequencies should be removed before to the DFB. This provides another reason to combine the DFB with a multiresolution scheme.

One way of achieving a multiscale decomposition is to use a Laplacian pyramid (LP) introduced by Burt and Adelson [18]. The LP decomposition at each step generates a sampled lowpass version of the original and the difference between the original and the prediction, resulting in a bandpass image. A drawback of the LP is the implicit oversampling. However, in contrast to the critically sampled wavelet scheme, the LP has the distinguishing feature that each pyramid level generates only one bandpass image (even for multidimensional cases) which does not have "scrambled" frequencies. As illustrated in Figure 2.2, this frequency scrambling happens in the wavelet filter bank when a highpass channel, after downsampling, is folded back into the low frequency band, and thus its spectrum is reflected. In the LP, this effect is avoided by downsampling the lowpass channel only.

Therefore the LP permits further subband decomposition to be applied on its bandpass images. Those bandpass images can be fed into a DFB so that directional information can be well captured. Figure 4.4 depicts this pyramidal directional filter bank (PDFB). The scheme can be iterated repeatedly on the coarse image. The end result is a decomposition into directional subbands at multiple scales. The scheme is flexible since it allows for a different number of directions at each scale.



(a) (b)

Figure 4.4: Pyramidal directional filter bank. (a) Block diagram. First, a standard multiscale decomposition into octave bands, where the lowpass channel is subsampled while the highpass is not. Then, a directional decomposition by the DFB is applied to each highpass channel. (b) Resulting frequency division, where the number of directions is increased with frequency.

With perfect reconstruction LP and DFB, the PDFB is obviously perfect reconstruction, and thus it is a frame operator for 2-D signals. The PDFB has

the same redundancy as the LP: up to 33% when subsampling by two in each dimension. If the filters in the LP are orthogonal (that is, $h[\boldsymbol{n}] = g[-\boldsymbol{n}]$ and $g[\boldsymbol{n}]$ is orthogonal to its translates with respect to the subsampling lattice), then it can be shown that LP is a tight frame with energy conservation [56]. In that case, assuming an image x is decomposed into J bandpass images $b_j,\ j = 1, 2, \ldots, J$ and a lowpass image a_J,[2] then we have

$$\|x\|^2 = \sum_{j=1}^{J} \|b_j\|^2 + \|a_J\|^2. \tag{4.13}$$

In addition, we assume that the DFB is also orthonormal, which means that it transforms each bandpass image b_j into a set of coefficients d_j with: $\|b_j\|^2 = \|d_j\|^2$. Then the decomposition by PDFB: $x \mapsto (d_1, d_2, \ldots, d_J, a_J)$ has also the energy conservation property:

$$\|x\|^2 = \sum_{j=1}^{J} \|d_j\|^2 + \|a_J\|^2. \tag{4.14}$$

Thus we have the following result on the PDFB.

Proposition 4.1 *The PDFB is a tight frame with frame bounds equal to* 1 *when orthogonal filters are used in both the LP and the DFB.*

Tight frames have important properties in some applications. For example, the error introduced in the transform domain is the same as the error in the reconstructed signal.

Let us point out that there are other multiscale and directional decompositions such as the cortex transform [178] and the steerable pyramid [152]. Our PDFB differs from those in that it allows different number of directions at each scale while nearly achieving critical sampling. In addition, we make the link to continuous-time construction precise, both through a relation to curvelets, and by studying the convergence of iterated DFB's.

4.3.2 PDFB for Curvelets

Next we will demonstrate that a PDFB where *the number of directions is doubled at every* ***other*** *finer scale in the pyramid* satisfies these key properties of curvelets discussed in Section 4.2.1. That is, we apply a DFB with $\lfloor n_0 - j/2 \rfloor$ levels or $2^{\lfloor n_0 - j/2 \rfloor}$ direction to the bandpass image b_j of the LP. Thus, the PDFB provides a discrete implementation for the curvelet transform.

A LP, with downsampling by two in each direction, is taken at every level, providing an octave-band decomposition: the LP bandpass image b_j at the level j creates subband with a corona support based on the interval $[\pi 2^{-j}, \pi 2^{-j+1}]$, for $j = 1, 2, \ldots, J$. Combining this with a directional decomposition by a DFB, we obtain the frequency tiling for curvelets as shown in Figure 4.5.

[2] The index is such that the level $j = 1$ corresponds to the finest scale.

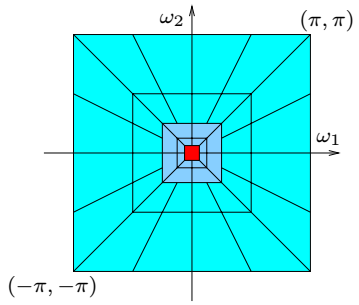


Figure 4.5: Resulting frequency division by a pyramidal directional filter bank for the curvelet transform. As the scale is refined from coarse to fine, the number of directions is doubled at every **other** octave band.

In term of basis functions, a coefficient in the LP subband b_j corresponds to a basis function that has local support in a square of size about 2^j. While a basis function from a DFB with $\lfloor n_0 - j/2 \rfloor$ iterated levels has support in a rectangle of length about $2^{n_0 - j/2}$ and width about 1. Therefore, in the PDFB, a basis function at the pyramid level j has support as:

$$width \approx 2^j \quad \text{and} \quad length \approx 2^j . 2^{n_0 - j/2} = 2^{n_0} 2^{j/2}, \tag{4.15}$$

which clearly satisfies the anisotropy scaling relation (4.1) of curvelets.

Figure 4.6 graphically depicts this property of a PDFB implementing a curvelet transform. As can be seen from the two shown pyramidal levels, the support size of the LP is reduced by four times while the number of directions of the DFB is doubled. With this, the support size of the PDFB basis images are changed from one level to next in accordance with the curve scaling relation. Also note that in this representation, as the scale is getting finer, there are more directions.

Figure 4.7 shows some real basis images from an actual PDFB that implements the digital curvelet transform. Again we can see that as the scale getting coarser, these basis images have their widths double at every level, while their lengths double at every other level. Therefore, in overall these basis images satisfy the anisotropy scaling relation.

Figure 4.8 shows an example image that is transformed by the PDFB implementing the discrete ridgelet transform. As we can see, the coefficients in the transform domain are very sparse – significant coefficients are located around edges *and* in the right directional subbands. With non-linear approximation using the PDFB, smooth regions are represented efficiently by the small size lowpass image while smooth edges are efficiently represented by a few directional local coefficients.

In [159], Starck et al. describe a different approach for the digital curvelet transform, in which they directly "discretize" the continuous definition. Their implementation uses the discrete Radon transform on image blocks, and thus the number of represented directions, which equals the block size, is reduced by

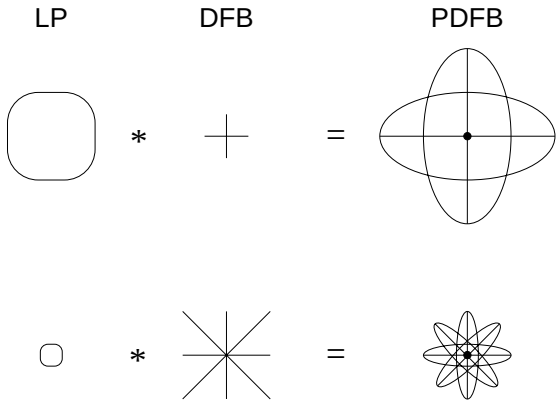


Figure 4.6: Illustration of the PDFB basis images for the curvelet transform. From the upper line to the lower line, the scale is reduced by four while the number of directions is doubled.

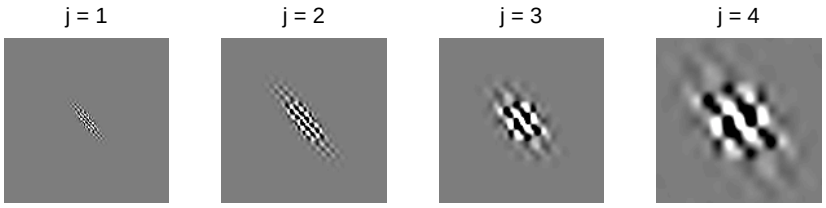
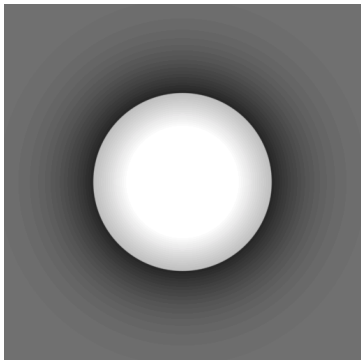


Figure 4.7: Basis images from a PDFB that implements the curvelet transform. They have linear structure and satisfy the anisotropy scaling relation.

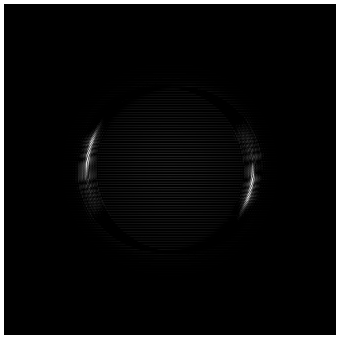
half for every finer scale. This is unlike the curvelet construction in continuous space, or our construction. Furthermore, there is a redundancy factor equals to $16J + 1$ in their implementation compared with 1.3 in ours.

4.4 Multiresolution Analysis

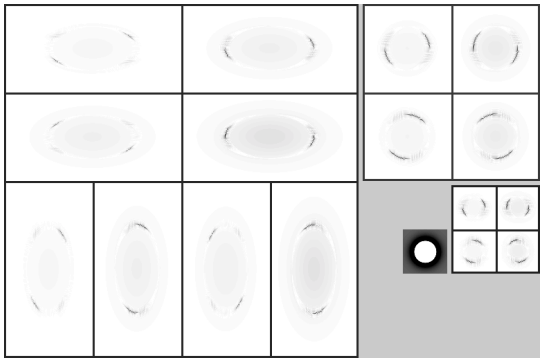
Since curvelets and PDFB are multiscale transforms, they exhibit successive approximation properties. The key in understanding this multiresolution analysis is the embedded grids of approximation [103]. The multiresolution analysis is the bridge that provides the link between the continuous domain construction (i.e. curvelets) and the discrete domain implementation (i.e. PDFB). The original curvelet construction of Candès and Donoho [20] uses block ridgelet transforms implies a mixture of rectangular and polar grids, and thus it is difficult to see how the embedded grids of approximation can be defined. Our filter bank approach to the curvelet transform makes this task much simpler. In fact, the sampling matrices from the PDFB for the curvelet transform directly result in a *rectangular* embedded grid system.



(a)



(c)



(b)

Figure 4.8: Example of PDFB. (a) Input image. (b) Magnitudes of PDFB coefficients. (c) Reconstruction from one PDFB subband. The LP uses the biorthogonal "9-7" filters while the DFB's use the biorthogonal "23-45" quincunx filters (mentioned in Section 3.3).

4.4.1 Multiscale

The discussion in Section 2.6.2, which we will briefly review, shows that associated with the discrete computation of the Laplacian pyramid is a multiscale system in the continuous-domain. Suppose that the LP in the PDFB uses orthogonal filters and downsampling by two is taken in each dimension. Under certain conditions, the lowpass filter G in the LP uniquely define an orthogonal scaling function $\phi(t) \in L^2(\mathbb{R}^2)$ via the two-scale equation

$$\phi(t) = 2 \sum_{n \in \mathbb{Z}^2} g[n] \phi(2t - n)$$

Denote

$$\phi_{j,n} = 2^{-j} \phi \left(\frac{t - 2^j n}{2^j} \right), \qquad j \in \mathbb{Z}, n \in \mathbb{Z}^2.$$

Then the family $\{\phi_{j,n}\}_{n\in\mathbb{Z}^2}$ is an orthonormal basis of V_j for all $j \in \mathbb{Z}$. The sequence of nested subspaces $\{V_j\}_{j\in\mathbb{Z}}$ satisfies the following invariant properties:

$$\begin{aligned} \text{Shift invariance:} \quad & f(t) \in V_j \quad \Leftrightarrow \quad f(t - 2^j k) \in V_j, \quad \forall j \in \mathbb{Z}, k \in \mathbb{Z}^2 \\ \text{Scale invariance:} \quad & f(t) \in V_j \quad \Leftrightarrow \quad f(2^{-1}t) \in V_{j+1}, \quad \forall j \in \mathbb{Z}. \end{aligned}$$

In other words, V_j is a subspace defined on a uniform grid with intervals $2^j \times 2^j$, which characterize the image approximation at the resolution 2^{-j}. The difference image in the LP carry the details necessary to increase the resolution of an image approximation. Let W_j be the orthogonal complement of V_j in V_{j-1} (also see Figure 4.9)

$$V_{j-1} = V_j \oplus W_j$$

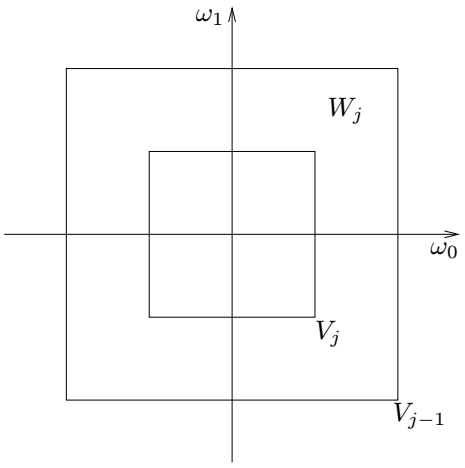


Figure 4.9: Multiscale subspaces generated by the Laplacian pyramid.

The oversampled filter bank view of the LP in Section 2.5 indicates that each polyphase component of the difference signal can be considered as a separate filter bank channel like the coarse signal. Let $F_i(z), 0 \le i \le 3$ be the synthesis filters for these polyphase components. Note that $F_i(z)$ are highpass filters. As in the wavelet filter bank, we associate with each of these filters a continuous functions $\psi^{(i)}(t)$ where

$$\psi^{(i)}(t) = 2 \sum_{n\in\mathbb{Z}^2} f_i[n]\phi(2t - n).$$

Theorem 2.5 states that for scale 2^j, $\{\psi^{(i)}_{j,n}\}_{0\le i\le 3,\ n\in\mathbb{Z}^2}$ is a tight frame of W_j. For all scales, $\{\psi^{(i)}_{j,n}\}_{0\le i\le 3,\ j\in\mathbb{Z},\ n\in\mathbb{Z}^2}$ is a tight frame of $L^2(\mathbb{R}^2)$. In all cases, the frame bounds are equal to 1.

Since W_{j+1} is generated by four prototype functions, in general it is *not* a shift invariant subspace, unless $F_i(z)$ are shifted versions of a filter, or

$$F_i(z) = z^{-k_i} F(z) \tag{4.16}$$

where k_i are the coset representatives of the downsampling lattice $(2,2)$

$$k_0 = (0,0)^T,\ k_1 = (1,0)^T,\ k_2 = (0,1)^T,\ \text{and}\ k_3 = (1,1)^T. \tag{4.17}$$

Nevertheless, based on this, we can mimic W_{j+1} to be a shift invariant subspace by denoting

$$\mu_{j,2n+k_i}(t) = \psi^{(i)}_{j+1,n} = \sum_{m\in\mathbb{Z}^2} f_i[m]\phi_{j,n+m}(t). \tag{4.18}$$

With this notation, the family $\{\mu_{j,n}\}_{n\in\mathbb{Z}^2}$ is a tight frame of W_{j+1} and it assimilates a uniform grid on $\mathbb{R}^2$ of intervals $2^j \times 2^j$.

4.4.2 Multidirection

Suppose that the DFB's in the PDFB use orthogonal filters. Results in Chapter 3 state that with a l-levels DFB, the family

$$\{g^{(l)}_k[\cdot - \mathsf{S}^{(l)}_k n]\}_{0\le k\le 2^l-1,\ n\in\mathbb{Z}^2} \tag{4.19}$$

is a directional orthonormal basis of $l^2(\mathbb{Z}^2)$. Here $g^{(l)}_k[n]$ are directional filters, in which $k = 0,\ldots,2^{l-1}-1$ correspond to the directions in $[-45°,+45°]$, while $k = 2^{l-1},\ldots,2^l-1$ correspond directions in $[+45°,+135°]$. The sampling matrices $\mathsf{S}^{(l)}_k$ are given in (3.23) which are all diagonal.

In the PDFB, the discrete basis (4.19) of the DFB can be regarded as a change of basis for the continuous subspaces from the multiscale decomposition. Although in the PDFB, the DFB is applied to the difference signal or the W_{j+1} subspaces, we will first study what happens when the DFB is applied to the multiresolution subspaces V_j.

Proposition 4.2 *Define*

$$\theta^{(l)}_{j,k,n}(t) = \sum_{m\in\mathbb{Z}^2} g^{(l)}_k[m - \mathsf{S}^{(l)}_k n]\phi_{j,m}(t) \tag{4.20}$$

The family $\{\theta^{(l)}_{j,k,n}\}_{n\in\mathbb{Z}^2}$ *is an orthonormal basis of a directional subspace* $V^{(l)}_{j,k}$ *for each* $k = 0,\ldots,2^l-1$. *These subspaces are orthogonal with*

$$V^{(l)}_{j,k} = V^{(l+1)}_{j,2k} \oplus V^{(l+1)}_{j,2k+1}, \quad \textit{and} \tag{4.21}$$

$$V_j = \bigoplus_{k=0}^{2^l-1} V^{(l)}_{j,k}. \tag{4.22}$$

Proof: (*Sketch*) This result is proved by induction on the number of decomposition levels l of the DFB, in much the same way for the wavelet packets bases [34] (see also [103]). Assume that $\{\theta^{(l)}_{j,k,n}\}_{n\in\mathbb{Z}^2}$ is an orthonormal basis of a subspace $V^{(l)}_{j,k}$. From Section 3.3, we see that to increase the directional resolution, an extra level of decomposition by a pair of orthogonal filters is

applied to the channel represented by $g_k^{(l)}$ that leads to two channels with equivalent filters $g_{2k}^{(l+1)}$ and $g_{2k+1}^{(l+1)}$. This transforms the orthonormal basis $\{\theta_{j,k,n}^{(l)}\}_{n\in\mathbb{Z}^2}$ in two orthonormal families $\{\theta_{j,2k,n}^{(l+1)}\}_{n\in\mathbb{Z}^2}$ and $\{\theta_{j,2k+1,n}^{(l+1)}\}_{n\in\mathbb{Z}^2}$. Each of these families generates a subspace with finer directional resolution that satisfy the "two-direction" equation (4.22). With this, starting from the orthonormal basis $\{\phi_{j,n}\}_{n\in\mathbb{Z}^2}$ of V_j, all other orthonormal bases follow. □

Figure 4.10 illustrates the "two-direction" subspace splitting by the DFB in the frequency domain. In the spatial domain, $V_{j,k}^{(l)}$ is a subspace at a scale 2^j and a direction k among all 2^l directions.

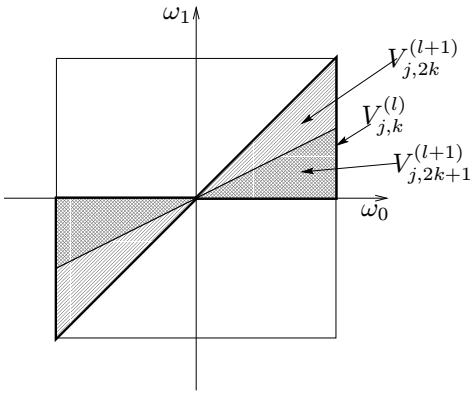


Figure 4.10: Multidirection subspaces generated by the DFB.

Next by applying the directional decomposition by the family (4.19) onto the detail subspace W_{j+1} as done by the PDFB, we obtain the similar result.

Proposition 4.3 *Define*

$$\rho_{j,k,n}^{(l)}(t) = \sum_{m\in\mathbb{Z}^2} g_k^{(l)}[m - \mathsf{S}_k^{(l)} n]\mu_{j,m}(t) \tag{4.23}$$

The family $\{\rho_{j,k,n}^{(l)}\}_{n\in\mathbb{Z}^2}$ *is a tight frame of a subspace* $W_{j+1,k}^{(l)}$ *with frame bounds equal to 1, for each* $k = 0, \ldots, 2^l - 1$. *These subspaces are orthogonal with*

$$W_{j+1,k}^{(l)} = W_{j,2k}^{(l+1)} \oplus W_{j,2k+1}^{(l+1)}, \quad \textit{and} \tag{4.24}$$

$$W_{j+1} = \bigoplus_{k=0}^{2^l-1} W_{j+1,k}^{(l)}. \tag{4.25}$$

Proof: This result is obtained by applying Theorem 2.5 to the subspaces in Proposition 4.2. □

Figure 4.11 shows a graphical representation of the subspaces in Proposition 4.3, seen in the frequency domain. The reason for $\{\rho^{(l)}_{j,k,n}\}_{n\in\mathbb{Z}^2}$ to be an overcomplete system for $W^{(l)}_{j+1,k}$ is because it uses the same sampling grid as the bigger subspace $V^{(l)}_{j,k}$.

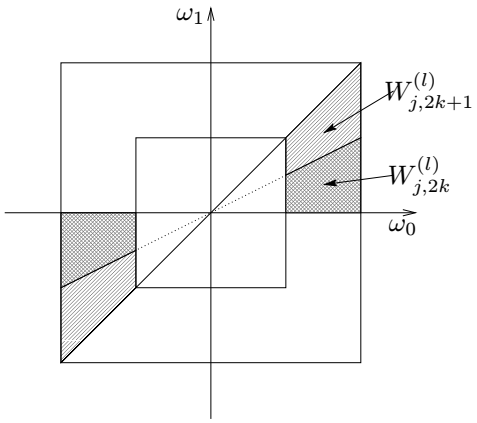


Figure 4.11: Multidirection subspaces generated by the PDFB.

Recall that W_{j+1} is *not* shift invariant but the following result establishes that its subspaces $W^{(l)}_{j+1,k}$ are since they are generated by a single prototype function.

Proposition 4.4 *Let us denote*

$$\rho^{(l)}_{j,k}(t) = \sum_{m\in\mathbb{Z}^2} g^{(l)}_k[m]\mu_{j,m}(t) \tag{4.26}$$

Then for $l \geq 2$

$$\rho^{(l)}_{j,k,n}(t) = \rho^{(l)}_{j,k}(t - 2^j \mathsf{S}^{(l)}_k n) \tag{4.27}$$

Proof: The definition of $\psi_{j,n}(t)$ in (2.54) implies that

$$\psi_{j,m+n}(t) = \psi_{j,m}(t - 2^j n).$$

Applying this to (4.18) we have

$$\mu_{j,m+2n}(t) = \mu_{j,m}(t - 2^j 2n).$$

In other words, $\mu_{j,m}$ are periodically shift invariant with even shifts in $m \in \mathbb{Z}^2$. From (3.17) we see that with $l \geq 2$, sampling by $\mathsf{S}^{(l)}_k$ is also even in each dimension. Thus, with a change of variable we obtain

$$\begin{aligned}
\rho^{(l)}_{j,k,n}(t) &= \sum_{m\in\mathbb{Z}^2} g^{(l)}_k[m - \mathsf{S}^{(l)}_k n]\, \mu_{j,m}(t) \\
&= \sum_{m\in\mathbb{Z}^2} g^{(l)}_k[m]\, \mu_{j,m}(t - 2^j \mathsf{S}^{(l)}_k n) \\
&= \rho^{(l)}_{j,k}(t - 2^j \mathsf{S}^{(l)}_k n).
\end{aligned}$$

□

Consequently, the subspaces $W_{j+1,k}^{(l)}$ satisfy the following shift invariant property:

$$f(t) \in W_{j+1,k}^{(l)} \quad \Leftrightarrow \quad f(t - 2^j S_k^{(l)} n) \in W_{j+1,k}^{(l)}, \quad \forall n \in \mathbb{Z}^2. \tag{4.28}$$

This says that the directional multiscale subspaces $W_{j+1,k}^{(l)}$ are defined on a rectangular grid with intervals $2^{j+l-1} \times 2^{j+1}$ (or $2^{j+1} \times 2^{j+l-1}$, depending on whether it is basically horizontal or vertical). For later reference, by substituting (4.18) into (4.26) we can write the prototype function $\rho_{j,k}^{(l)}(t)$ directly as a linear combination of the scaling function $\phi_{j,m}(t)$ as

$$\begin{aligned} \rho_{j,k}^{(l)}(t) &= \sum_{i=0}^{3} \sum_{n} g_k^{(l)}[2n + k_i] \left(\sum_{m \in \mathbb{Z}^2} f_i[m] \phi_{j,n+m} \right) \\ &= \sum_{m \in \mathbb{Z}^2} \underbrace{\left(\sum_{i=0}^{3} \sum_{n \in \mathbb{Z}^2} g_k^{(l)}[2n + k_i] f_i[m - n] \right)}_{c_k^{(l)}[m]} \phi_{j,m}(t). \end{aligned} \tag{4.29}$$

The sequence $c_k^{(l)}[m]$ resemble a summation of convolutions between $g_k^{(l)}[m]$ and $f_i[m]$, thus it is a highpass and directional filter. Equation (4.29) reveals the "ridgelet-like" behavior of the prototype function $\rho_{j,k}^{(l)}(t)$ for $W_{j+1,k}$. Moreover, we see that $\rho_{j,k}^{(l)}(t)$ resembles a grouping of "edge-detection" elements at a scale indexed by j and along a direction indexed by k.

4.4.3 Multiscale and Multidirection

Finally, integrating over scales we have the following result for the frames on the space $L^2(\mathbb{R}^2)$.

Theorem 4.1 *For a sequence of finite positive integers $\{l_j\}_{j \le j_0}$ the family*

$$\{\phi_{j_0,n}(t),\ \rho_{j,k,n}^{(l_j)}(t)\}_{j \le j_0,\ 0 \le k \le 2^{l_j}-1,\ n \in \mathbb{Z}^2} \tag{4.30}$$

is a tight frame of $L^2(\mathbb{R}^2)$. For a sequence of finite positive integers $\{l_j\}_{j \in \mathbb{Z}}$, the family

$$\{\rho_{j,k,n}^{(l_j)}(t)\}_{j \in \mathbb{Z},\ 0 \le k \le 2^{l_j}-1,\ n \in \mathbb{Z}^2} \tag{4.31}$$

is a directional wavelet tight frame of $L^2(\mathbb{R}^2)$. In each case, the frame bounds are equal to 1.

Proof: This result is obtained by applying Proposition 4.3 to the following decompositions of $L^2(\mathbb{R}^2)$ into mutual orthogonal subspaces:

$$\begin{aligned} L^2(\mathbb{R}^2) &= V_{j_0} \oplus \left(\bigoplus_{j \le j_0} W_j \right), \quad \text{and} \\ L^2(\mathbb{R}^2) &= \bigoplus_{j \in \mathbb{Z}} W_j. \end{aligned}$$

□

As discussed in Section 4.3.2, the tight frame in (4.30) provides a curvelet-like expansion when the number of directions is doubled at every other finer scale. This means that if at the scale 2^{j_0} we start with an l_{j_0}-level DFB (which has $2^{l_{j_0}}$ directions) then at finer scales 2^j, $j < j_0$, the number of decomposition levels by the DFB should be:

$$l_j = \lfloor l_{j_0} - (j - j_0)/2 \rfloor, \quad \text{for } j \le j_0. \tag{4.32}$$

Thus the embedded grid of approximation for the curvelet PDFB expansion at the scale 2^j is $2^{\lfloor n_0 + j/2 \rfloor} \times 2^j$ for basically horizontal directions and $2^j \times 2^{\lfloor n_0 + j/2 \rfloor}$ for near vertical directions, where $n_0 = l_{j_0} - j_0/2 + 2$. Figure 4.12 illustrates this sampling pattern at different scales and directions. The main point to note here is that in the refinement process, one spatial dimension is refined at twice the speed as the other spatial dimension.

Figure 4.6 and Figure 4.12 give a complete view of the multiresolution approximation of our curvelet construction based on the PDFB. They clearly show a refinement scheme where the resolution increases in both spatial and direction domain when going from coarse to fine scale, and the basis elements exhibit the anisotropy scaling relation for curves.

To conform with the wavelet terminology, in the family (4.30) that provides a curvelet expansion, the functions ϕ are called scaling functions, while the functions ρ are called curvelet functions.

4.5 Experimental Results

4.5.1 Non-linear Approximation

In this section, we evaluate the non-linear approximation performance of the PDFB that implements the discrete curvelet transform on some real images and compare it with the performance by the 2-D discrete wavelet transform (DWT2) The four test images shown in Figure 4.13 are: "Lena", "Barbara", "Peppers", and "D15" (a texture pattern from the Brodatz collection) – all of size 512×512. The first three images are piecewise smooth, while the last one is a directional texture.

The wavelet transform used in the experiment is a biorthogonal transform with the "9-7" filters [31] (see also Table 2.1) and 6 decomposition levels. The

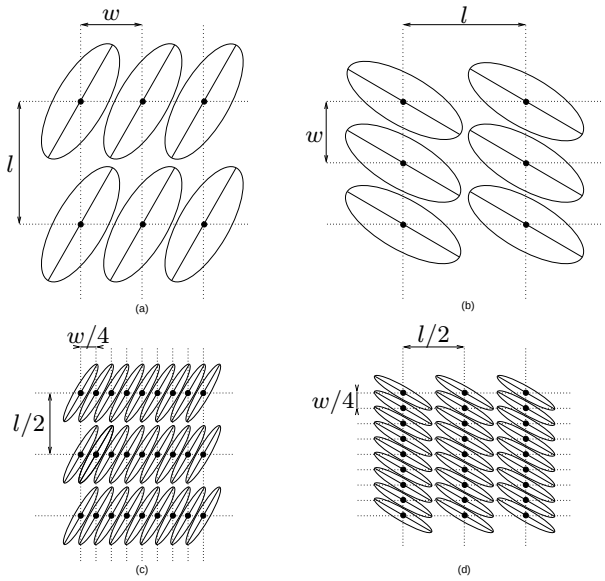


Figure 4.12: Embedded grids of approximation in spatial domain for a PDFB that implements the curvelet transform. These are four illustrative subspaces $W_{j,k}$ representing coarser vs. finer scales and basically horizontal vs. basically vertical directions. Each subspace is spanned by the shifts of a curvelet prototype function given in (4.26). The sampling intervals match with the supports of the prototype function, for example width w and length l, so that the shifts would tile the $\mathbb{R}^2$ plane.

PDFB also uses the "9-7" filters in the LP decomposition. While the DFB in the PDFB uses the "23-45" biorthogonal quincunx filters designed by Phoong et al. [129] that was mentioned in Section 3.3. The number of decomposition levels by the DFB at the finest pyramidal scale is 5, which leads to 32 directions.

Note that in this case, both the DWT2 and the PDFB transforms share the same multiscale detailed subspaces W_j as defined in Section 4.4.1, which are generated by the "9-7" lowpass filters. The difference is that in the DWT2, each subspace W_j is represented by a basis with three directions, whereas in the PDFB it is represented by a redundant frame with many more directions.

First, the NLA performance is evaluated in terms of PSNR for each transform as follows.

- For a given value M, select the M-most significant coefficients in transform domain.
- Reconstruct the image from these M coefficients and calculate the PSNR (peak signal-to-noise-ratio).
- Plot the PSNR's curve as a function of M.

Figure 4.14 shows the PSNR comparison results. For the "Lena" and "Pep-

(a) Lena

(b) Barbara

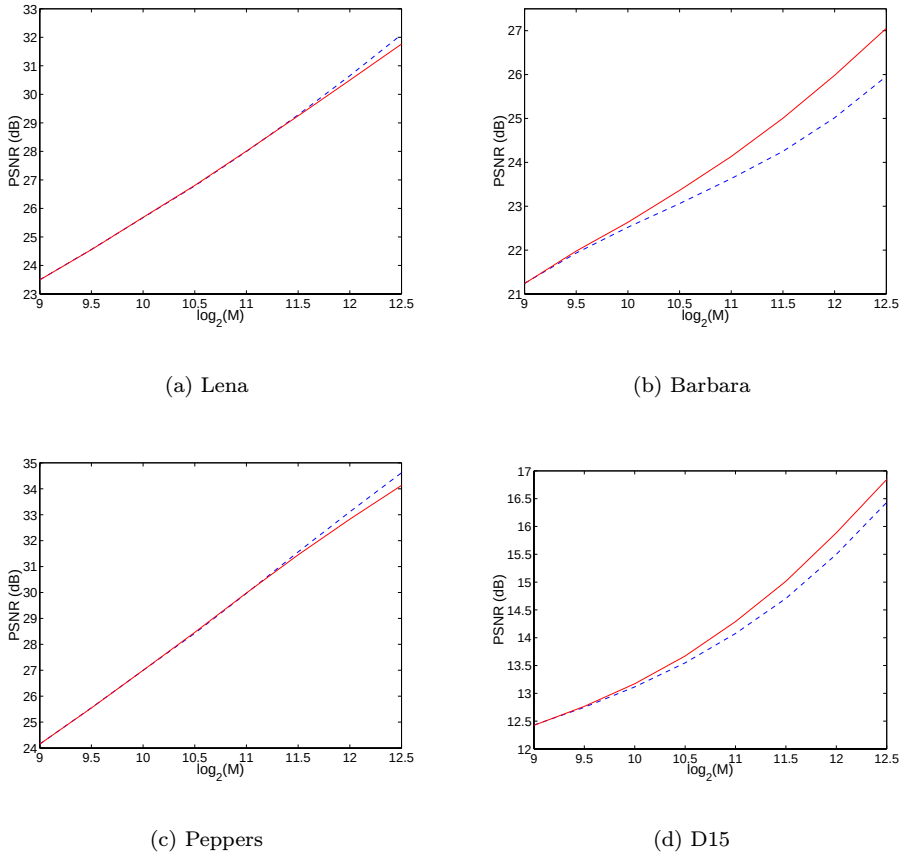
(c) Peppers

(d) D15

Figure 4.13: Four test images that are used in the experiments. They are 8-bit images of size 512×512.

pers" images, the two transforms have similar performance, while for "Barbara" and "D15", the PDFB slightly improves the PSNR.

The similar performance in the reconstruction error leads us to examine in detail the refinement schemes by the two transforms. Since the two transforms share the same detailed subspaces, it is possible to restrict the comparison in these subspaces. We expect most of the refinement actions would happen around the image edges. Figure 4.15 and Figure 4.16 show sequences of non-linear approximated images at the finest subspace W_j using the DWT2 and the PDFB, respectively. What we see clearly resembles the painting scenario that was discussed in the Introduction chapter. The DWT2 scheme slowly refines the



(a) Lena (b) Barbara

(c) Peppers (d) D15

Figure 4.14: Non-linear approximation results on different test images. DWT2: dashed lines. PDFB: solid lines.

detailed image by isolated "dots" along the contour, while the PDFB scheme quickly refines by well-adapted "sketches". The improvement by the PDFB can be seen both in terms of visual quality and the reconstruction error.

However, we observe that for typically smooth images, the energy distribution in the high frequency bands is relatively small compared to the low frequency bands. Therefore, the improvement in the detailed subspaces has little effect on the overall reconstruction error. This is reason why the PDFB did not give a significant improvement in terms of PSNR for the "Peppers" and "Lena" images. Nevertheless, the PDFB is promising with the visual quality (especially recovering image contours), as well as in image feature detection.

On the other hand, for the "Barbara" and "D15" images, where there is a significant energy distribution in the high frequency region, the PDFB gives a visible improvement. Figure 4.17 shows a detailed comparison of two non-linear approximated images by the DWT2 and the PDFB. In this case, fine contours (directional textures on cloths) are better represented by the PDFB transform.

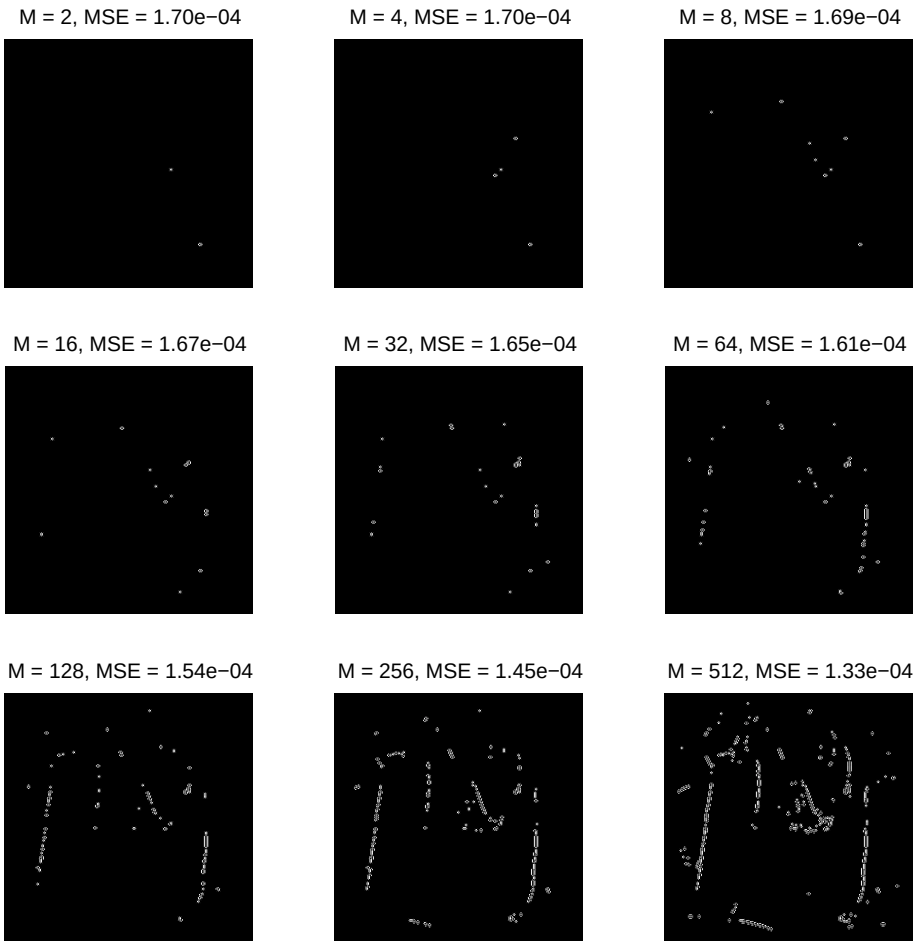


Figure 4.15: Sequence of images showing the non-linear approximation at the finest scale of the DWT2. M is the number of the most significant coefficients; MSE is the mean square error against the projection of the input image into the finest detailed subspace. The input is the "Peppers" image.

4.5.2 Image Denoising

The non-linear approximation power of the PDFB is furthermore tested in denoising experiments and compared with the wavelet transform. In both cases, a simple hard thresholding by the same threshold value is performed in the transform domain. In this comparison, the PDFB has a slight advantage over the DWT2 since the PDFB is oversampled with a small factor. Nevertheless, we want to verify the ability in recovering smooth edges by the PDFB representation. Figure 4.18 displays a detail comparison of a denoising experiment, where the PDFB is shown to be more effective in recovering edges, as well as in signal-to-noise ratio (SNR).

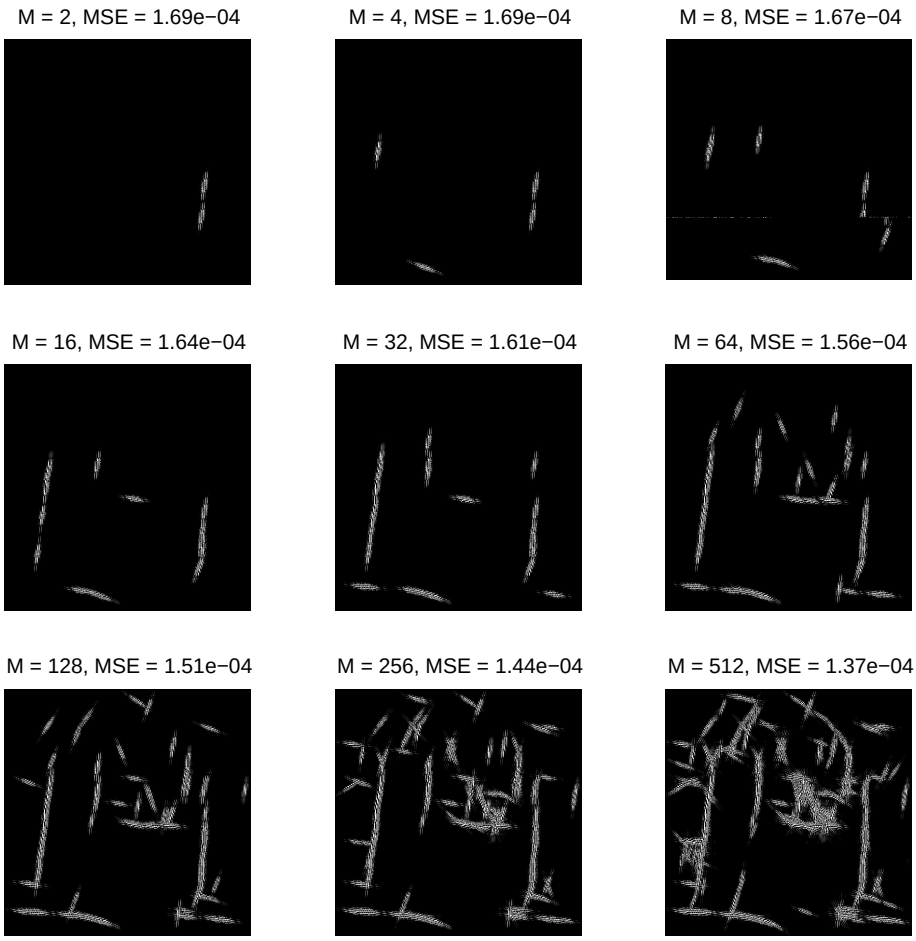


Figure 4.16: Same as in Figure 4.15 but with the PDFB. Note that the PDFB shares the same detailed subspace with the DWT2.

4.6 Conclusion

In this chapter we investigated the possibility of implementing the discrete curvelet transform that offers a sparse representation for piecewise smooth images using the filter bank approach. We first identified two key features of the curvelet transform that could lead to an improvement over the wavelet transform, namely directionality and anisotropy. From this we proposed a new filter bank structure, the pyramidal directional filter bank (PDFB), that can provide a curvelet-like decomposition for images with a small redundancy factor. The connection between curvelets and the PDFB was made precisely via a new directional multiresolution analysis, which provides successive refinements at *both* spatial and directional resolution using frame elements satisfying the anisotropy scaling relation for curves. Experiments with real images indicate the potential of the PDFB in image processing applications.

(a) Original image

(b) DWT2: PSNR = 24.34 dB

(c) PDFB: PSNR = 25.70 dB

Figure 4.17: Detail of non-linear approximated images by the DWT2 and the PDFB. In each case, the image originally of size 512×512 is reconstructed from the 4096-most significant coefficients in the transform domain.

We highlight below the advantages of the PDFB.

- It presents a efficient scheme for representing piecewise smooth images.
- It is a fixed transform, with effective tree data structures that can be explored in coding application.
- It has potential in other image processing applications such as feature detection and image analysis.

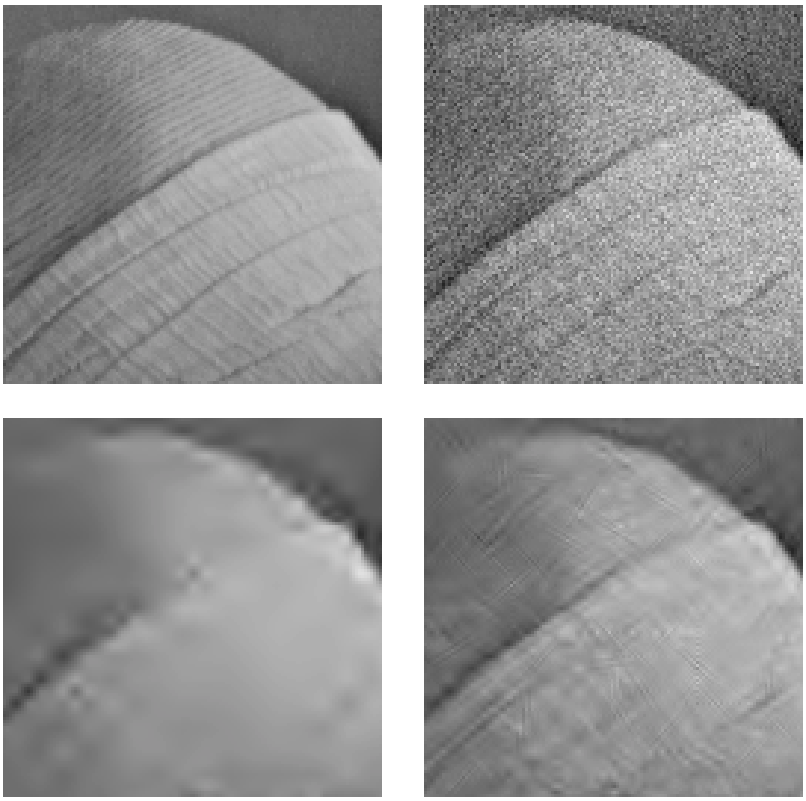


Figure 4.18: Denoising experiment: original image (top left), noisy image (top right, SNR = 9.55 dB), denoising using DWT2 (bottom left, SNR = 13.82 dB), and denoising using PDFB (bottom right, SNR = 15.42 dB).

Chapter 5

The Finite Ridgelet Transform

5.1 Introduction

Many image processing tasks take advantage of *sparse* representations of image data where most information is packed into a small number of samples. Typically, these representations are achieved via invertible and non-redundant transforms. Currently, the most popular choices for this purpose are the wavelet transform [175, 65, 103] and the discrete cosine transform [137].

To overcome the weakness of wavelets in higher dimensions, Candès and Donoho [19, 21] recently pioneered a new system of representations named *ridgelets* which deal effectively with *line* singularities in 2-D. The idea is to map a line singularity into a point singularity using the Radon transform [45]. Then, the wavelet transform can be used to effectively handle the point singularity in the Radon domain (Figure 5.1). Their initial proposal was intended for functions defined in the *continuous* $\mathbb{R}^2$ space.

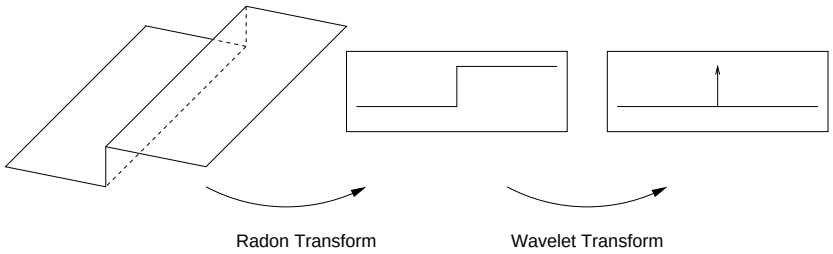


Figure 5.1: The ridgelet transform as the Radon transform followed by the wavelet transform.

For practical applications, the development of *discrete* versions of the ridgelet transform that lead to algorithmic implementations is a challenging problem. Due to the radial nature of ridgelets, straightforward implementations based on discretization of continuous formulae would require interpolation in polar

[0]This chapter includes research conducted jointly with Martin Vetterli [52, 51, 55].

coordinates, and thus result in transforms that would be either redundant or can not be perfectly reconstructed.

In [64, 159, 6], the authors take the redundant approach in defining discrete Radon transforms that can lead to invertible discrete ridgelet transforms with some appealing properties. For example, a recent preprint [6] proposes a new notion of Radon transform for data in a rectangular coordinate such that the lines exhibit geometrical faithfulness. Their transform is invertible with a factor four oversampled. However, the inverse transform is ill-conditioned in the presence of noise and requires an iterative approximation algorithm.

In this chapter, we propose a discrete ridgelet transform that achieves both invertibility and non-redundancy. In fact, our construction leads to a large family of *orthonormal* and directional bases for digital images, including adaptive schemes. As a result, the inverse transform is numerically stable and uses the same algorithm as the forward transform. Because a basic building block in our construction is the finite Radon transform [15], which has a wrap-around (or aliased line) effect, our ridgelet transform is not geometrically faithful. The properties of the new transform are demonstrated and studied in several applications.

The outline of this chapter is as follows. In the next section we review the concept and motivation of ridgelets in the continuous domain. In Section 5.3, we introduce the finite Radon transform with a novel ordering of coefficients as a key step in our discrete ridgelet construction. The finite Radon transform is then studied within the theory of frames. The finite ridgelet transform is defined in Section 5.4, where the main result is a general family of orthonormal transforms for digital images. In Sections 5.5, we propose several variations on the initial design of the finite ridgelet transform. Numerical experiments are presented in Section 5.6, where the new transform is compared with the traditional ones, especially the wavelet transform. We conclude in Section 5.7 with some discussions.

5.2 Continuous Ridgelet Transform

We start by briefly reviewing the ridgelet transform and showing its connections with other transforms in the continuous domain. Given an integrable bivariate function $f(\boldsymbol{x})$, its *continuous ridgelet transform* (CRT) in $\mathbb{R}^2$ is defined by [19, 21]

$$CRT_f(a,b,\theta) = \int_{\mathbb{R}^2} \psi_{a,b,\theta}(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x}, \tag{5.1}$$

where the ridgelets $\psi_{a,b,\theta}(\boldsymbol{x})$ in 2-D are defined from a wavelet-type function in 1-D $\psi(x)$ as

$$\psi_{a,b,\theta}(\boldsymbol{x}) = a^{-1/2}\psi((x_1\cos\theta + x_2\sin\theta - b)/a). \tag{5.2}$$

Figure 5.2 shows a typical ridgelet: the function is oriented at an angle θ and is constant along the lines $x_1\cos\theta + x_2\sin\theta = const$.

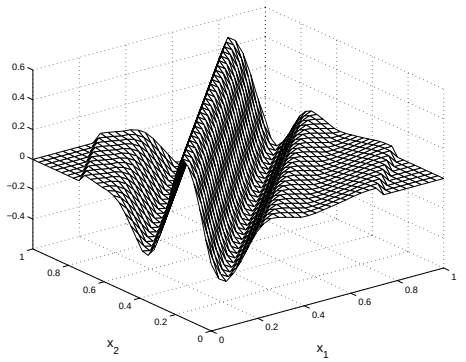


Figure 5.2: A typical ridgelet function $\psi_{a,b,\theta}(x_1, x_2)$.

For comparison, the (separable) continuous wavelet transform (CWT) in $\mathbb{R}^2$ of $f(\boldsymbol{x})$ can be written as

$$CWT_f(a_1, a_2, b_1, b_2) = \int_{\mathbb{R}^2} \psi_{a_1,a_2,b_1,b_2}(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x}, \tag{5.3}$$

where the wavelets in 2-D are tensor products

$$\psi_{a_1,a_2,b_1,b_2}(\boldsymbol{x}) = \psi_{a_1,b_1}(x_1)\psi_{a_2,b_2}(x_2), \tag{5.4}$$

of 1-D wavelets, $\psi_{a,b}(t) = a^{-1/2}\psi((t-b)/a)$.[1]

As can be seen, the CRT is similar to the 2-D CWT except that the *point* parameters (b_1, b_2) are replaced by the *line* parameters (b, θ). In other words, these 2-D multiscale transforms are related by:

Wavelets: $\rightarrow$ $\psi_{scale,\ point-position}$,
Ridgelets: $\rightarrow$ $\psi_{scale,\ line-position}$.

As a consequence, wavelets are very effective in representing objects with isolated point singularities, while ridgelets are very effective in representing objects with singularities along lines. In fact, one can think of ridgelets as a way of concatenating 1-D wavelets along lines. Hence the motivation for using ridgelets in image processing tasks is appealing since singularities are often joined together along edges or contours in images.

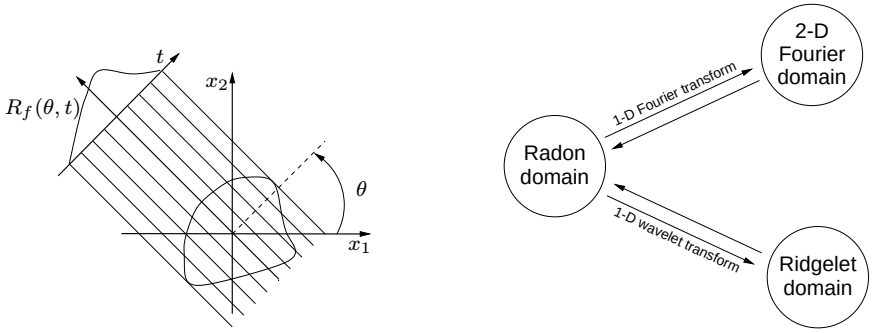
In 2-D, points and lines are related via the Radon transform, thus the wavelet and ridgelet transforms are linked via the Radon transform. More precisely, denote the Radon transform as

$$R_f(\theta, t) = \int_{\mathbb{R}^2} f(\boldsymbol{x})\delta(x_1 \cos\theta + x_2 \sin\theta - t) d\boldsymbol{x}, \tag{5.5}$$

[1] In practice, however one typically enforces the same dilation scale on both directions thus leading to three wavelets corresponding to horizontal, vertical and diagonal directions.

then the ridgelet transform is the application of a 1-D wavelet transform to the slices (also referred to as projections) of the Radon transform,

$$CRT_f(a, b, \theta) = \int_{\mathbb{R}} \psi_{a,b}(t) R_f(\theta, t) dt. \tag{5.6}$$



(a) (b)

Figure 5.3: Radon, Fourier and ridgelet transform. (a) The Radon transform in $\mathbb{R}^2$. For a fixed θ, $R_f(\theta, t)$ is a slice or projection in the Radon domain. (b) Relations between transforms. The ridgelet transform is the application of 1-D wavelet transform to the slices of the Radon transform, while the 2-D Fourier transform is the application of 1-D Fourier transform to those Radon slices.

Figure 5.3(a) shows a graphical representation of the Radon transform. It is instructive to note that if in (5.6) instead of taking a 1-D wavelet transform, the application of a 1-D Fourier transform along t would result in the 2-D Fourier transform. More specifically, let $F_f(\boldsymbol{\omega})$ be the 2-D Fourier transform of $f(\boldsymbol{x})$, then we have

$$F_f(\xi \cos\theta, \xi \sin\theta) = \int_{\mathbb{R}} e^{-j\xi t} R_f(\theta, t) dt. \tag{5.7}$$

This is the famous *projection-slice* theorem and is commonly used in image reconstruction from projection methods [79, 142]. The relations between the various transforms are shown in Figure 5.3(b).

5.3 Finite Radon Transform

5.3.1 Forward and Inverse Transforms

As suggested in the previous section, a discrete ridgelet transform can be obtained via a discrete Radon transform. Numerous discretizations of the Radon transforms have been devised to approximate the continuous formulae [110, 79, 142, 12, 145, 89]. However, most of them were not designed to be

invertible transforms for digital images. Alternatively, the *finite* Radon transform theory (which means transform for *finite* length signals) [15, 71, 108, 144] originated from combinatorics, provides an interesting solution. Also, in [177], a closely related transform is derived from the periodization of the continuous Radon transform.

The *finite Radon transform* (FRAT) is defined as summations of image pixels over a certain set of "lines". Those lines are defined in a finite geometry in a similar way as the lines for the continuous Radon transform in the Euclidean geometry. Denote $Z_p = \{0, 1, \ldots, p-1\}$, where p is a prime number. Note that Z_p is a finite field with modulo p operations [98]. For later convenience, we denote $Z_p^* = \{0, 1, \ldots, p\}$.

The FRAT of a real function f on the finite grid Z_p^2 is defined as

$$r_k[l] = FRAT_f(k,l) = \frac{1}{\sqrt{p}} \sum_{(i,j)\in L_{k,l}} f[i,j]. \tag{5.8}$$

Here $L_{k,l}$ denotes the set of points that make up a line on the lattice Z_p^2, or more precisely

$$\begin{aligned} L_{k,l} &= \{(i,j) : j = ki + l \pmod p,\ i \in Z_p\}, \quad 0 \le k < p, \\ L_{p,l} &= \{(l,j) : j \in Z_p\}. \end{aligned} \tag{5.9}$$

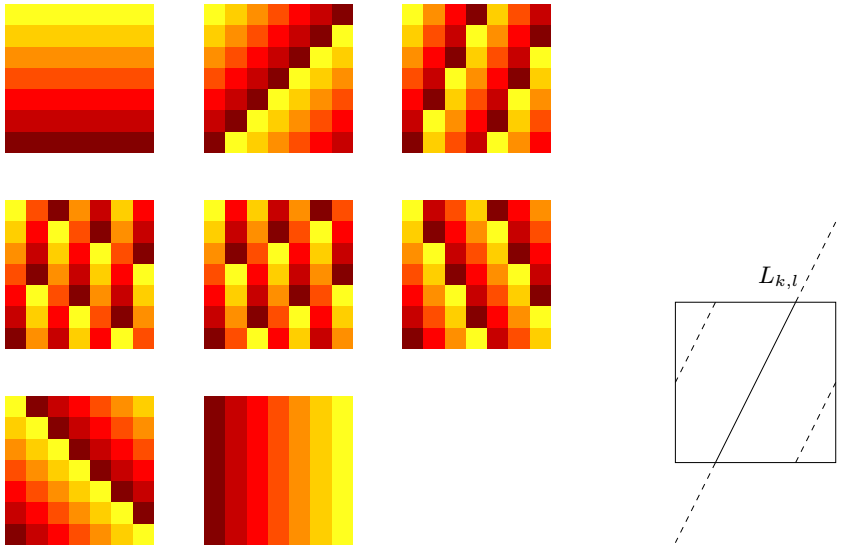
Figure 5.4 shows an example of the finite lines $L_{k,l}$ where points in the grid Z_p^2 are represented by image pixels. Note that due to the modulo operations in the definition of lines for the FRAT, these lines exhibit a "wrap around" effect. In other words, the FRAT treat the input image as one period of a periodic image. Later, we will present several ways to limit this artifact.

We observe that in the FRAT domain, the energy is best compacted if the mean is subtracted from the image $f[i,j]$ prior to taking the transform given in (5.8), which is assumed in the sequel. We also introduce the factor $p^{-1/2}$ in order to normalize the l_2-norm between the input and output of the FRAT.

Just as in the Euclidean geometry, a line $L_{k,l}$ on the affine plane Z_p^2 is uniquely represented by its slope or direction $k \in Z_p^*$ ($k = p$ corresponds to infinite slope or vertical lines) and its intercept $l \in Z_p$. One can verify that there are $p^2 + p$ lines defined in this way and every line contains p points. Moreover, any two distinct points on Z_p^2 belong to just one line. Also, two lines of different slopes intersect at exactly one point. For any given slope, there are p parallel lines that provide a complete cover of the plane Z_p^2. This means that for an input image $f[i,j]$ with zero-mean, we have

$$\sum_{l=0}^{p-1} r_k[l] = \frac{1}{\sqrt{p}} \sum_{(i,j)\in Z_p^2} f[i,j] = 0 \quad \forall k \in Z_p^*. \tag{5.10}$$

Thus, (5.10) explicitly reveals the redundancy of the FRAT: in each direction, there are only $p-1$ independent FRAT coefficients. Those coefficients at $p+1$ directions together with the mean value make up totally of



(a) (b)

Figure 5.4: (a) Lines for the 7×7 FRAT. Parallel lines are grouped in each of the eight possible directions. Images in order from top to bottom, left to right are corresponding to the values of k from 0 to 7. In each image, points (or pixels) in different lines are assigned with different gray-scales. (b) "Wrap around" effect as a result of the modulo operations in the definition of lines for the FRAT

$(p+1)(p-1)+1 = p^2$ independent coefficients (or degrees of freedom) in the finite Radon domain, as expected.

By analogy with the continuous case, the *finite back-projection* (FBP) operator is defined as the sum of Radon coefficients of all the lines that go through a given point, that is

$$FBP_r(i,j) = \frac{1}{\sqrt{p}} \sum_{(k,l) \in P_{i,j}} r_k[l], \quad (i,j) \in Z_p^2, \tag{5.11}$$

where $P_{i,j}$ denotes the set of indices of all the lines that go through a point $(i,j) \in Z_p^2$. More specifically, using (5.9) we can write

$$P_{i,j} = \{(k,l) : l = j - ki \pmod p, k \in Z_p\} \cup \{(p,i)\}. \tag{5.12}$$

From the property of the finite geometry Z_p^2 that every two points lie on exactly one line, it follows that every point in Z_p^2 lies on exactly one line from the set $P_{i,j}$, except for the point (i,j) which lies on all $p+1$ lines. Thus, by

substituting (5.8) into (5.11) we obtain

$$\begin{aligned} FBP_r(i,j) &= \frac{1}{p} \sum_{(k,l)\in P_{i,j}} \sum_{(i',j')\in L_{k,l}} f[i',j'] \\ &= \frac{1}{p} \left(\sum_{(i',j')\in Z_p^2} f[i',j'] + p.f[i,j] \right) \\ &= f[i,j]. \end{aligned} \tag{5.13}$$

So the back-projection operator defined in (5.11) indeed computes the inverse FRAT for zero-mean images. Therefore we have an efficient and exact reconstruction algorithm for the FRAT. Furthermore, since the FBP operator is the adjoint of the FRAT operator, the algorithm for the inverse of FRAT has the same structure and is symmetric with the algorithm for the forward transform.

It is easy to see that the FRAT requires *exactly* p^3 additions and p^2 multiplications. Moreover, for memory access efficiency, [108] describes an algorithm for the FRAT in which for each projection k we need to pass through every pixel of the original image only once using p histogrammers, one for each summation in (5.8) of that projection. For images of moderate sizes, we observed that the actual computational time of the FRAT is compatible with other $O(p^2 \log(p^2)$ transforms, such as the 2-D FFT, where the leading constant can be large. For example, on a Sun Ultra 5 computer, both the forward and inverse FRAT's take less than a second to compute on an image of size 257×257.

5.3.2 Optimal Ordering of the Finite Radon Transform Coefficients

The FRAT described in Section 5.3.1 uses (5.9) as a convenient way of specifying finite lines on the Z_p^2 grid via two parameters: the slope k and the intercept l. However it is neither a unique nor the best way for our purpose. Let us consider a more general definition of lines on the finite Z_p^2 plane as

$$L'_{a,b,t} = \left\{ (i,j) \in Z_p^2 : ai + bj - t = 0 \pmod p \right\}, \tag{5.14}$$

where $a, b, t \in Z_p$ and $(a,b) \neq (0,0)$.

This is by analogy with the line equation: $x_1 \cos\theta + x_2 \sin\theta - t = 0$ in $\mathbb{R}^2$. Therefore, for a finite line defined as in (5.14), (a,b) has the role of the normal vector, while t is the translation parameter. In this section, all equations involving line parameters are carried out in the finite field Z_p, which is assumed in the sequel without the indication of mod p.

To relate (5.9) with (5.14), consider the general line equation in (5.14)

$$ai + bj - t = 0. \tag{5.15}$$

If $b \neq 0$ then, (5.15) $\Leftrightarrow j = -b^{-1}ai + b^{-1}t$, where b^{-1} denotes the multiplicative inverse of b in the finite field Z_p, i.e. $bb^{-1} = 1 \pmod p$. Otherwise, if $b = 0$, then it is necessary that $a \neq 0$, thus (5.15) $\Leftrightarrow i = a^{-1}t$.

By matching the line equations in (5.9) and (5.14), we have the following equivalences between these two specifications of finite lines in Z_p^2:

$$\begin{aligned} L'_{a,b,t} &\equiv L_{k,l} \quad \text{if } a = -kb, b \neq 0, t = bl, \quad \text{for } 0 \leq k < p, \quad \text{and} \\ L'_{a,b,t} &\equiv L_{p,l} \quad \text{if } a \neq 0, b = 0, t = al. \end{aligned}$$

In other words, there is a many to one (precisely, $p-1$ to one) mapping between the line parameters in (5.14) and in (5.9), such that they represent the same line. It is easy to see that for any $c \in Z_p, c \neq 0$ then $\{cl : l \in Z_p\}$ is the same as the set Z_p. For a fixed normal vector (a, b), the set of parallel lines $\left\{L'_{a,b,t} : t \in Z_p\right\}$ equals to the set of p lines $\{L_{k,l} : l \in Z_p\}$ with the same slope k, where $k = -b^{-1}a$ for $b \neq 0$ and $k = p$ for $b = 0$. Moreover, the set of lines with the normal vector (a, b) is also equal to the set of lines with the normal vector (na, nb), for each $n = 1, 2, \ldots, p-1$.

With the general line specification in (5.14), we now define the new FRAT to be

$$r_{a,b}[t] = FRAT_f(a, b, t) = \frac{1}{\sqrt{p}} \sum_{(i,j) \in L'_{a,b,t}} f[i, j]. \tag{5.16}$$

From the discussion above we see that a new FRAT projection sequence: $(r_{a,b}[0], r_{a,b}[1], \ldots, r_{a,b}[p-1])$, is simply a reordering of a projection sequence $(r_k[0], r_k[1], \ldots, r_k[p-1])$ from (5.8). This ordering is important for us since we later apply a 1-D wavelet transform on each FRAT projection. Clearly, the chosen normal vectors (a, b) control the order for the coefficients in each FRAT's projection, as well as the represented directions of those projections.

The usual FRAT described in Section 5.3.1 uses the set of $(p+1)$ normal vectors $\boldsymbol{u}_k$, where

$$\begin{aligned} \boldsymbol{u}_k &= (-k, 1) \quad \text{for } k = 0, 1, \ldots, p-1, \text{ and} \\ \boldsymbol{u}_p &= (1, 0). \end{aligned} \tag{5.17}$$

In order to provide a complete representation, we need the FRAT to be defined as in (5.16) with a set of $p+1$ normal vectors $\left\{(a_k, b_k) : k \in Z_p^*\right\}$ such that they cover all $p+1$ distinct FRAT projections represented by $\left\{\boldsymbol{u}_k : k \in Z_p^*\right\}$. We have $p-1$ choices for each of those normal vectors as

$$(a_k, b_k) = n\boldsymbol{u}_k, \quad 1 \leq n \leq p-1.$$

So what is the good choice for the $p+1$ normal vectors of the FRAT? To answer this we first prove the following projection slice theorem for the general FRAT. A special case of this theorem is already shown in [108].

Defining $W_p = e^{-2\sqrt{-1}\pi/p}$, the *discrete Fourier transform* (DFT) of a function f on Z_p^2 can be written as

$$F[u, v] = \frac{1}{p} \sum_{(i,j) \in Z_p^2} f[i, j] W_p^{ui+vj}, \tag{5.18}$$

and for FRAT projections on Z_p as

$$R_{a,b}[w] = \frac{1}{\sqrt{p}} \sum_{t \in Z_p} r_{a,b}[t] W_p^{wt}. \tag{5.19}$$

Theorem 5.1 (Discrete projection-slice theorem) *The 1-D DFT $R_{a,b}[w]$ of a FRAT projection $r_{a,b}[t]$ is identical to the 2-D DFT $F[u,v]$ of $f[i,j]$ evaluated along a discrete slice through the origin at direction (a,b):*

$$R_{a,b}[w] = F[aw, bw]. \tag{5.20}$$

Proof: Substituting (5.16) into (5.19) and using the fact that the set of p parallel lines $\left\{L'_{a,b,t} : t \in Z_p\right\}$ provides a complete cover of the plane Z_p^2, we obtain

$$\begin{aligned} R_{a,b}[w] &= \frac{1}{p} \sum_{t \in Z_p} \sum_{(i,j) \in L'_{a,b,t}} f[i,j] W_p^{wt} \\ &= \frac{1}{p} \sum_{(i,j) \in Z_p^2} f[i,j] W_p^{w(ai+bj)} \\ &= F[aw, bw]. \end{aligned}$$

□

From (5.20), we can see the role of the FRAT normal vectors (a,b) in the DFT domain: it controls the order of the coefficients in the corresponding Fourier slices. In particular, $F[a,b]$ equals to the first harmonic component of the FRAT projection sequence with the normal vector (a,b). For the type of images that we are interested in, e.g. of natural scenes, most of the energy is concentrated in the low frequencies. Therefore in these cases, in order to ensure that each FRAT projection is smooth or low frequency dominated so that it can be represented well by the wavelet transform, the represented normal vector (a,b) should be chosen to be as "close" to the origin of the Fourier plane as possible.

Figure 5.5 illustrates this by showing an example of a discrete Fourier slice. The normal vector for the corresponding FRAT projection can be chosen as a vector from the origin to *any* other point on the Fourier slice. However, the best normal vector is selected as the closest point to the origin. The choice of the normal vector (a,b) as the closest point to the origin causes the represented direction of the FRAT projection to have the least "wrap around" due to the periodization of the transform. The effect of the new ordering of FRAT coefficient in the image domain is illustrated in Figure 5.6 for the same example projection. As can be seen, the "wrap around" effect is significantly reduced with the optimal ordering compared to the usual one.

Formally, we define the set of $p+1$ optimal normal vectors $\left\{(a_k^\star, b_k^\star) : k \in Z_p^*\right\}$ as follows

$$(a_k^\star, b_k^\star) = \arg \min_{\substack{(a_k,b_k) \in \{n\mathbf{u}_k : 1 \le n \le p-1\} \\ \text{s.t. } \mathcal{C}_p(b_k) \ge 0}} \|(\mathcal{C}_p(a_k), \mathcal{C}_p(b_k))\|. \tag{5.21}$$

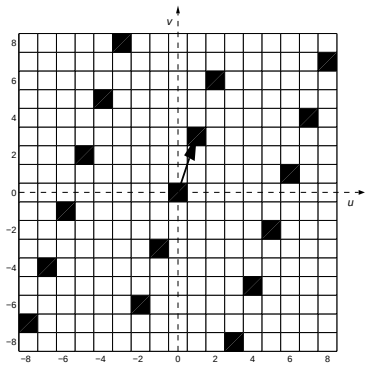


Figure 5.5: Example of a discrete Fourier slice (indicated by the black squares) with the best normal vector for that FRAT projection. In this example, $p = 17$ and the slope $k = 11$. The normal vector can be chosen as a vector from the origin to any other points on the Fourier slide. The best normal vector is $(1, 3)$ (the solid arrow).

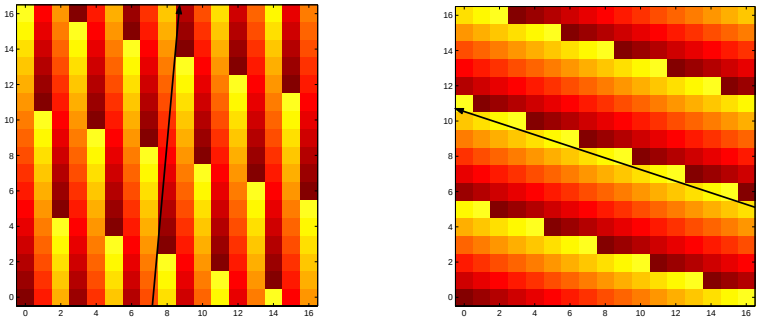
Here $\mathcal{C}_p(x)$ denotes the centralized function of period p: $\mathcal{C}_p(x) = x - p.\text{round}(x/p)$. Hence, $\|(\mathcal{C}_p(a_k), \mathcal{C}_p(b_k))\|$ represents the distance from the origin to the point (a_k, b_k) on the periodic Fourier plane as shown in Figure 5.5. The constraint $\mathcal{C}_p(b_k) \geq 0$ is imposed in order to remove the ambiguity in deciding between (a, b) and $(-a, -b)$ as the normal vector for a projection. As a result, the optimal normal vectors are restricted to have angles in $[0, \pi)$. We use norm-2 for solving (5.21). Minimization is simply done for each $k \in Z_p^*$ by computing $p - 1$ distances in (5.21) and select the smallest one. Figure 5.7 shows an example of the optimal set of normal vectors. In comparison with the usual set of normal vectors $\left\{\boldsymbol{u}_k : k \in Z_p^*\right\}$ as given in (5.17), the new set $\left\{(a_k^\star, b_k^\star) : k \in Z_p^*\right\}$ provides a much more uniform angular coverage.

After obtaining the set of normal vectors $\{(a_k^\star, b_k^\star)\}$, we can compute the FRAT and its inverse with the same fast algorithms using histogrammers described in Section 5.3.1. For a given p, solving (5.21) requires $O(p^2)$ operations and therefore it is negligible compared to the transforms themselves. Furthermore, this can be pre-computed, thus only presents as a "one-time" cost.

For the sake of simplicity, we write $r_k[t]$ for $r_{a_k^\star, b_k^\star}[t]$ in the sequel. In other words, from now we regard k as an index in the set of optimal FRAT normal vectors rather than a slope. Likewise, the line $L'_{a_k^\star, b_k^\star, t}$ is simply rewritten as $L_{k,t}$, for $0 \leq k \leq p,\ 0 \leq t < p$.

5.3.3 Frame Analysis of the FRAT

Since the FRAT is a redundant transform, it can be studied as a *frame operator*. In this section we will study the FRAT in more detail and reveal some of its properties in this frame setting. A brief introduction to frames was given in Section 2.2.4. For finite dimensional frames, suppose that $\mathcal{F}$ is a linear operator



(a) (b)

Figure 5.6: Lines for the FRAT projection as shown in Figure 5.5 using: (a) usual ordering, (b) optimal ordering. They both represent the same set of lines but with different orderings. The orderings are signified by the increasing of gray-scales. The arrows indicate the represented directions in each case.

from $\mathbb{R}^N$ to $\mathbb{R}^M$, defined by

$$(\mathcal{F}x)_n = \langle x, \varphi_n \rangle, \qquad \text{for } n = 1, \dots, M.$$

The set $\{\varphi_n\}_{n=1}^M \subset \mathbb{R}^N$ is called a frame of $\mathbb{R}^N$ if the frame condition (2.11) is satisfied, and in that case, $\mathcal{F}$ is called a frame operator. It can be shown that any finite set of vectors that spans $\mathbb{R}^N$ is a frame. The frame bound ratio B/A indicates the numerical stability in reconstructing x from $(\mathcal{F}x)_n$; the tighter the frame, the more stable the reconstruction against coefficient noise.

The frame operator can be regarded as a left matrix multiplication with F, where F is an $M \times N$ matrix in which its nth row equals to φ_n. The frame condition (2.11) can be rewritten as

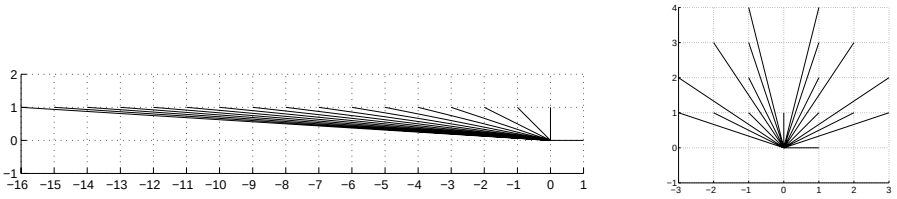
$$x^T A x \leq x^T F^T F x \leq x^T B x, \qquad \forall x \in \mathbb{R}^N. \tag{5.22}$$

Since $F^T F$ is symmetric, it is diagonalizable in an orthonormal basis [80], thus (5.22) implies that the eigenvalues of $F^T F$ are between A and B. Therefore, the tightest possible frame bounds A and B are the minimum and maximum eigenvalues of $F^T F$, respectively. In particular, a tight frame is equivalent to $F^T F = A \cdot I$, which means the transpose of F equals to its left inverse within a scale factor A.

Now let us return to the FRAT. Since it is invertible it can be regarded as a frame operator in $l_2(Z_p^2)$ with the frame $\left\{\varphi_{k,l} : k \in Z_p^*, l \in Z_p\right\}$ defined as

$$\varphi_{k,l} = p^{-1/2} \delta_{L_{k,l}} \tag{5.23}$$

where δ_S denotes the characteristic function for the set S, which means $\delta_S[i,j]$ equals to 1 if $(i,j) \in S$ and 0 otherwise.



(a) (b)

Figure 5.7: The set of normal vectors, which indicate the represented directions, for the FRAT of size $p = 17$ using: (a) usual ordering; (b) optimal ordering.

Note that this frame is normalized since $\|\varphi_{(k,l)}\| = 1$. Moreover, using properties of lines in the finite geometry Z_p^2, it is easy to verify that

$$\langle \varphi_{k,l}, \varphi_{k',l'} \rangle = \begin{cases} 1 & \text{if } k = k', l = l' \\ 0 & \text{if } k = k', l \neq l' \\ 1/p & \text{if } k \neq k' \end{cases} \tag{5.24}$$

Hence, the minimum angle between any two frame vectors of the FRAT is: $\cos^{-1}(1/p)$, which approaches the right angle for large p. Therefore, we can state that the FRAT frame is "almost" orthogonal.

By writing images as column vectors, the FRAT can be regarded as a left matrix multiplication with $F = p^{-1/2}R$, where $\{R\}_{(k,l),\ (i,j)}$ is the $(p^2+p) \times p^2$ incidence matrix of the affine geometry Z_p^2: $R_{(k,l),\ (i,j)}$ equals to 1 if $(i,j) \in L_{k,l}$ and 0 otherwise.

Proposition 5.1 *The tightest bounds for the FRAT frame* $\left\{\varphi_{k,l} : k \in Z_p^*, l \in Z_p\right\}$ *in* $l_2(Z_p^2)$ *are* $A = 1$ *and* $B = p + 1$.

Proof: From (5.22), these tightest bounds can be computed from the eigenvalues of $C = F^T F = p^{-1} R^T R$. Since R is the incidence matrix for lines in Z_p^2, $(R^T R)_{(i,j),\ (i',j')}$ equals the number of lines that go through both (i,j) and (i',j'). Using the properties of the finite geometry Z_p^2 that every two points lie in exactly one line and that there are exactly $p+1$ lines that go through each point, it follows that the entries of C equal to $(p+1)p^{-1}$ along its diagonal and p^{-1} elsewhere.

The key observation is that C is a circulant matrix, hence its eigenvalues can be computed as the p^2-points discrete Fourier transform (DFT) on its first column $c = ((p+1)p^{-1}, p^{-1}, \ldots, p^{-1})$ [175] (§2.4.8). Writing c as

$$c = (1, 0, \ldots, 0) + p^{-1} \cdot (1, 1, \ldots, 1),$$

we obtain,

$$DFT\{c\} = (1, 1, \ldots, 1) + p \cdot (1, 0, 0, \ldots, 0) = (p+1, 1, 1, \ldots, 1)$$

where the DFT is computed for the Dirac and constant signals.

Therefore the eigenvalues of C are $p+1$ and 1, the latter with multiplicity of p^2-1. As a result, the tightest (normalized) frame bounds for FRAT as $A=1$ and $B=p+1$. □

For reconstruction, the FBP defined in (5.11) can be represented by a left multiplication with matrix $p^{-1/2}B$, where $B_{(i,j),\ (k,l)}$ equals to 1 if $(k,l) \in P_{i,j}$ and 0 otherwise. From the definition of $P_{i,j}$, we have

$$R_{(k,l),\ (i,j)} = B_{(i,j),\ (k,l)}, \quad \forall i,j,k,l.$$

So the transform matrices for the operators FRAT and FBP are transposed of each other. Let $\bar{Z}_p^2$ denotes the subspace of zero-mean images defined on Z_p^2. Since the FBP is an inverse of the FRAT for zero-mean images, we have the following result.

Proposition 5.2 *On the subspace of zero-mean images $\bar{Z}_p^2$, the FRAT is a* normalized tight frame *with $A=B=1$, which means*

$$f = \sum_{k=0}^{p}\sum_{l=0}^{p-1}\langle f, \varphi_{k,l}\rangle \varphi_{k,l}, \quad \forall f \in \bar{Z}_p^2. \tag{5.25}$$

Remark 5.1 *It is worthwhile to note that constant images on Z_p^2 are eigenvectors of $C=F^TF$ with the eigenvalue $p+1$. Taking constant images out leaves a system with all unity eigenvalues, or a tight frame on the remaining subspace. Thus, another interpretation of FRAT is that it is a normalized tight frame for zero-mean images.*

By subtracting the mean from the image before applying the FRAT, we change the frame bound ratio from $p+1$ to 1 and obtain a tight frame. Consequently, this makes the reconstruction more robust against noise on the FRAT coefficients due to thresholding and/or quantization. This follows from the result in [72] that with the additive white noise model for the coefficients, the tight frame is optimal among normalized frames in minimizing mean-squared error.

5.4 Orthonormal Finite Ridgelet Transform

With an invertible FRAT and applying (5.6), we can obtain an invertible discrete ridgelet transform by taking the discrete wavelet transform (DWT) on each FRAT projection sequence, $(r_k[0], r_k[1], \ldots, r_k[p-1])$, where the direction k is fixed. We call the overall result the *finite ridgelet transform* (FRIT). Figure 5.8 depicts these steps.

Typically p is not dyadic, therefore a special border handling is required. Appendix 5.A details one possible way of computing the DWT for prime length

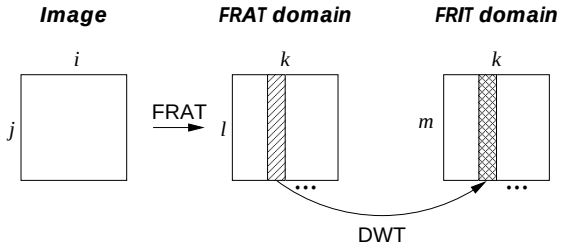


Figure 5.8: Diagram for the FRIT. After taking the FRAT, a DWT is applied on each of the FRAT slices or projections where k is fixed.

signals. Due to the periodicity property of the FRAT coefficients for each direction, periodic wavelet transforms are chosen and assumed in this section.

Recall that the FRAT is redundant and not orthogonal. Next we will show that by taking the 1-D DWT on the projections of the FRAT in a special way, we can remove this redundancy and obtain an orthonormal transform.

Assume that the DWT is implemented by an orthogonal tree-structured filter bank with J levels, where G_0 and G_1 are low and high pass synthesis filters, respectively. Then the family of functions:

$$\left\{ g_0^{(J)}[\cdot - 2^J m],\ g_1^{(j)}[\cdot - 2^j m] : j = 1, \ldots, J;\ m \in \mathbb{Z} \right\}$$

is the orthogonal basis of the discrete-time wavelet series [175]. Here $G^{(j)}$ denotes the equivalent synthesis filters at level j, or more specifically

$$\begin{aligned} G_0^{(J)}(z) &= \prod_{k=0}^{J-1} G_0(z^{2^k}), \\ G_1^{(j)}(z) &= G_1(z^{2^{j-1}}) \prod_{k=0}^{j-2} G_0(z^{2^k}), \quad j = 1, \ldots, J. \end{aligned}$$

The basis functions from $G_0^{(J)}$ are called the scaling functions, while all the others functions in the wavelet basis are called wavelet functions. Typically, the filter G_1 is designed to satisfy the high pass condition, $G_1(z)|_{z=1} = 0$ so that the corresponding wavelet has at least one vanishing moment. Therefore, $G_1^{(j)}(z)|_{z=1} = 0$, $\forall j = 1, \ldots, J$, which means all wavelet basis functions have zero mean.

For a more general setting, let us assume that we have a collection of $(p+1)$ 1-D orthonormal transforms on $\mathbb{R}^p$ (which can be the same), one for each projection k of FRAT, that have bases as

$$\left\{ \boldsymbol{w}_m^{(k)} : m \in Z_p \right\}, \quad k = 0, 1, \ldots, p.$$

The only condition that we require for each of these bases can be expressed equivalently by the following lemma.

Lemma 5.1 (Condition Z) *Suppose that* $\{\boldsymbol{w}_m : m \in Z_p\}$ *is an orthogonal basis for the finite-dimensional space* $\mathbb{R}^p$*, then the following are equivalent:*

1. *This basis contains a constant function, say* $\boldsymbol{w}_0$*, i.e.* $w_0[l] = const,\ \forall l \in Z_p$.

2. *All other basis functions,* $\boldsymbol{w}_m,\ m = 1, \ldots, p-1$*, have zero mean.*

Proof: Denote $\mathbf{1} = (1, 1, \ldots, 1) \in \mathbb{R}^p$. If $\boldsymbol{w}_0 = c\mathbf{1},\ c \neq 0$ then from the orthogonality assumption that $\langle \boldsymbol{w}_0, \boldsymbol{w}_m \rangle = 0$, we obtain $\sum_l w_m[l] = 0,\ \forall m = 1, \ldots, p-1$.

Conversely, assume that each basis function $\boldsymbol{w}_m,\ 1 \leq m \leq p-1$, has zero mean. Denote S the subspace that is spanned by these functions and $S^\perp$ is its orthogonal complement subspace in $\mathbb{R}^p$. It is clear that $S^\perp$ has dimension 1 with $\boldsymbol{w}_0$ as its basis. Consider the subspace $S_0 = \{c\mathbf{1} : c \in \mathbb{R}\}$. We have $\langle c\mathbf{1}, \boldsymbol{w}_m \rangle = c \sum_l w_m[l] = 0,\ \forall m = 1, \ldots, p-1$, thus $S_0 \subset S^\perp$. On the other hand, $\dim(S_0) = \dim(S^\perp) = 1$, therefore $S^\perp = S_0$. This means $\boldsymbol{w}_0$ is a constant function. □

As shown before, the *Condition Z* is satisfied for all wavelet bases, or in fact any general tree-structured filter banks where the all-lowpass branch is carried to the maximum number of stages (i.e. when only one scaling coefficient is left).

By definition, the FRIT can be written as

$$\begin{aligned} FRIT_f[k,m] &= \langle FRAT_f[k,\cdot], w_m^{(k)}[\cdot] \rangle \\ &= \sum_{l \in Z_p} w_m^{(k)}[l] \langle f, \varphi_{k,l} \rangle \\ &= \langle f, \sum_{l \in Z_p} w_m^{(k)}[l]\ \varphi_{k,l} \rangle. \end{aligned} \tag{5.26}$$

Here $\{\varphi_{k,l}\}$ is the FRAT frame which is defined in (5.23). Hence we can write the basis functions for the FRIT as follows:

$$\rho_{k,m} = \sum_{l \in Z_p} w_m^{(k)}[l]\ \varphi_{k,l}. \tag{5.27}$$

We can next prove the result on the orthogonality of a modified FRIT.

Theorem 5.2 *Given* $p+1$ *orthonormal bases in* $l^2(Z_p)$ *(which can be the same):* $\left\{ \boldsymbol{w}_m^{(k)} : m \in Z_p \right\},\ 0 \leq k \leq p$*, that satisfy the* Condition Z *then*

$$\{\rho_{k,m} :\ k = 0, 1, \ldots, p;\ m = 1, 2, \ldots, p-1\} \cup \{\rho_0\}$$

is an orthonormal basis in $l^2(Z_p^2)$*, where* $\rho_{k,m}$ *are defined in (5.27) and* ρ_0 *is the constant function,* $\rho_0[i,j] = 1/p,\ \forall (i,j) \in Z_p^2$.

Proof: Let us consider the inner products between any two FRIT basis functions

$$\langle \rho_{k,m}, \rho_{k',m'} \rangle = \sum_{l,l' \in Z_p} w_m^{(k)}[l]\ w_{m'}^{(k')}[l']\ \langle \varphi_{k,l}, \varphi_{k',l'} \rangle.$$

Using (5.24), when the two FRIT basis functions have the same direction, $k = k'$, then

$$\langle \rho_{k,m}, \rho_{k,m'} \rangle = \sum_{l \in Z_p} w_m^{(k)}[l]\ w_{m'}^{(k)}[l] = \delta[m - m'].$$

So the orthogonality of these FRIT basis functions comes from the orthogonality of the basis $\left\{ \boldsymbol{w}^{(k)} : m \in Z_p \right\}$. In particular, we see that $\rho_{k,m}$ have unit norm. Next, for the case when the two FRIT basis functions have different directions, $k \neq k'$, again by using (5.24) we obtain

$$\langle \rho_{k,m}, \rho_{k',m'} \rangle = \frac{1}{p} \sum_{l,l' \in Z_p} w_m^{(k)}[l]\ w_{m'}^{(k')}[l'] = \frac{1}{p} \left(\sum_{l \in Z_p} w_m^{(k)}[l] \right) \left(\sum_{l' \in Z_p} w_{m'}^{(k')}[l'] \right).$$

In this case, if either m or m' is non-zero, e.g. $m \neq 0$, then using the *Condition Z* of these bases, $\sum_{l \in Z_p} w_m^{(k)}[l] = 0$, it implies $\langle \rho_{k,m}, \rho_{k',m'} \rangle = 0$.

Finally, note that $\bigcup_l L_k(l) = Z_p^2$, for all directions k (see (5.10)). So, together with the assumption that $\boldsymbol{w}_0^{(k)}$ are constant functions, we see that all of the FRIT basis functions $\rho_{k,0}$, $(k = 0, 1, \ldots, p)$ correspond to the mean of the input image so we only need to keep one of them (in any direction), which is denoted as ρ_0. The proof is now complete. □

Remark 5.2 *1. An intuition behind the above result is that at each level of the DWT decomposition applied on the FRAT projections, all of the non-orthogonality and redundancy of the FRAT is pushed into the scaling coefficients. When the DWT's are taken to the maximum number of levels then all of the remaining scaling coefficients at different projections are the same, hence we can drop all but one of them. The result is an orthonormal FRIT.*

2. *We prove the above result for the general setting where different transforms can be applied on different FRAT projections. The choice of transforms can be either adaptive, depending on the image, or pre-defined. For example, one could employ an adaptive wavelet packet scheme independently on each projection. The orthogonality holds as long as the "all lowpass" branch of the general tree-structured filter bank is decomposed to a single coefficient. All other branches would contain at least one highpass filter thus leading to zero-mean basis functions.*

3. *Furthermore, due to the "wrap around" effect of the FRAT, some of its projections could contain strong periodic components so that a Fourier-type transform like the DCT might be more efficient. Also note that from*

Theorem 5.1, if we apply the 1-D Fourier transform on all of the FRAT projections then we obtain the 2-D Fourier transform. For convenience, we still use the term FRIT to refer to the cases where other transforms than the DWT might be applied to some of the FRAT projections.

To gain more insight into the construction for the orthogonal FRIT basis, Figure 5.9 illustrates a simple example of the transform on a 2×2 block using the Haar wavelet. In this case, the FRIT basis is the same as the 2-D Haar wavelet basis, as well as the 2-D discrete Fourier basis.

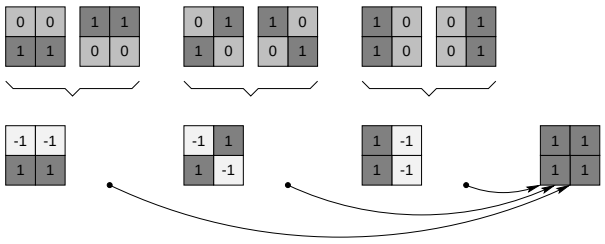


Figure 5.9: Illustration on the contraction of orthogonal FRIT basis for a 2×2 block using the Haar wavelet. *Upper:* Basis images for the FRAT. *Lower:* Basis images for the orthogonal FRIT. These images are obtained by taking the (scaled) Haar transform for each pair (corresponding to one projection) of the FRAT basis images. The constant image results from all projections and thus we can drop all but one of them.

5.5 Variations on the Theme

5.5.1 Folded FRAT and FRIT

The FRAT in the previous sections is defined with a periodic basis over Z_p^2. This is equivalent to applying the transform to a periodization of the input image f. Therefore relatively large amplitude FRAT coefficients could result due to the possible discontinuities across the image borders. To overcome this problem, we employ a similar strategy as in the block cosine transform by extending the image symmetrically about its borders [103].

Given that p is a prime number and $p > 2$, then p is odd and can be written as $p = 2n - 1$. Consider an $n \times n$ input image $f[i, j]$, $0 \leq i, j < n$. Fold this image with respect to the lines $i = 0$ and $j = 0$ to produce a $p \times p$ image $\check{f}[i, j]$, in which (also see Figure 5.10)

$$\check{f}[i, j] = f[|i|, |j|], \quad -n < i, j < n. \tag{5.28}$$

The periodization of $\check{f}[i, j]$ is symmetric and continuous across the borders of the original image, thus eliminating the jump discontinuity that would have resulted from the periodic extension of $f[i, j]$. Applying the FRAT to the $\check{f}[i, j]$ results in $p(p + 1)$ transform coefficients. Notice the new range for the pixel

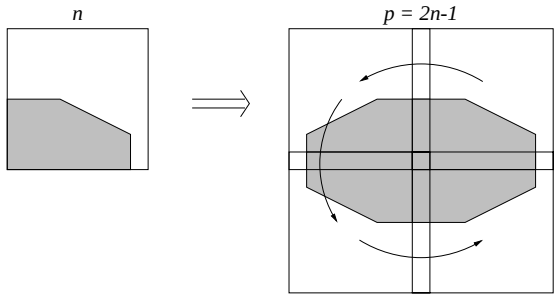


Figure 5.10: Extending the image symmetrically about its borders in order to reduce the discontinuities across the image borders due to the periodization.

indices of the image $\check{f}[i,j]$. We will show that the FRAT coefficients of $\check{f}[i,j]$ exhibit certain symmetry properties so that the original image can be perfectly reconstructed by keeping exactly n^2 coefficients.

Consider the 2-D DFT of $\check{f}[i,j]$

$$\check{F}[u,v] = \frac{1}{p} \sum_{-n<i,j<n} \check{f}[i,j] W_p^{ui+vj}.$$

Using the symmetry property of $\check{f}[i,j]$ in (5.28), we obtain

$$\check{F}[u,v] = \check{F}[|u|,|v|].$$

Theorem 5.1 shows that the FRAT $\check{r}_{a,b}[t],\ (-n < t < n)$ of $\check{f}[i,j]$ can be computed from the inverse 1-D DFT as

$$\check{r}_{a,b}[t] = \frac{1}{\sqrt{p}} \sum_{-n<w<n} \check{R}_{a,b}[w] W_p^{-wt},$$

where $\check{R}_{a,b}[w] = \check{F}[aw, bw]$. The symmetry of $\check{F}[u,v]$ thus yields

$$\check{R}_{a,b}[w] = \check{R}_{a,b}[|w|] \qquad \text{and} \tag{5.29}$$

$$\check{R}_{a,b}[w] = \check{R}_{|a|,|b|}[w]. \tag{5.30}$$

From (5.29) we have $\check{r}_{a,b}[t] = \check{r}_{a,b}[|t|]$ or each projection $\check{r}_{a,b}[t]$ is symmetric about $t = 0$, and (5.30) reveals the duplications among those projections. In fact, with the set of optimal normal vectors in (5.21), except for two projections indexed by $(1,0)$ and $(0,1)$ (the vertical and horizontal projections, respectively) all other projections have an identical twin. By removing those duplications we are left with $2 + (p-2)/2 = n+1$ projections. For example, we can select the set of $n+1$ independent projections as the ones with normal vectors in the first quadrant (refer to Figure 5.7). Furthermore, as in (5.10), the redundancy among the projections of the folded FRAT can be written as

$$\check{r}_{a_k^\star, b_k^\star}[0] + 2 \sum_{t=1}^{n-1} \check{r}_{a_k^\star, b_k^\star}[t] = \frac{1}{\sqrt{p}} \sum_{-n<i,j<n} \check{f}[i,j]. \tag{5.31}$$

The next proposition summarizes the above results.

Proposition 5.3 *The image $f[i,j]$ can be perfectly reconstructed from the following n^2-1 coefficients:*

$$\check{r}_{a_k^\star,b_k^\star}[t] \quad \textit{such that } \mathcal{C}_p(a_k^\star) \geq 0 \textit{ and } 0 < t < n, \tag{5.32}$$

and the mean of the image $\check{f}[i,j]$.

To gain better energy compaction, the mean should be subtracted from the image $\check{f}[i,j]$ previous to taking the FRAT. The set of independent coefficients in (5.32) is referred as the folded FRAT of the image $f[i,j]$.

However, orthogonality might be lost in the folded FRIT (resulting from applying 1-D DWT on $n+1$ projections of the folded FRAT), since the basis functions from a same direction of the folded FRAT could overlap. Nevertheless, if we loosen up the orthogonality constraint, then by construction, the folded FRAT projections $(\check{r}_{a_k^\star,b_k^\star}[t] : 0 < t < n)$ are symmetric with respect to $t = 0$ and $t = n - 1/2$. This allows the use of folded wavelet transform with biorthogonal symmetric wavelets [17] or orthogonal symmetric IIR wavelets [78].

5.5.2 Multilevel FRIT's

In the FRIT schemes described previously, multiscale comes from the 1-D DWT. As a result, at each scale, there is a large number of directions, which is about the size of the input image. Moreover, the basis images of the FRIT have long support, which extend over the whole image.

Here we describe a different scheme where the number of directions can be controlled, and the basis functions have smaller support. Assume that the input image has the size $n \times n$, where $n = p_1 p_2 \dots p_J q$ and p_i are prime numbers. First, we apply the orthonormal FRIT to $n_1 \times n_1$ non-overlapping subimages of size $p_1 \times p_1$, where $n_1 = p_2 \dots p_J q$. Each sub-image is transformed into $p_1^2 - 1$ "detail" FRIT coefficients plus a mean value. These mean values form an $n_1 \times n_1$ coarse approximate image of the original one. Then the process can be iterated on the coarse version up to J levels. The result is called as multilevel FRIT (MFRIT).

At each level, the basis functions for the "detail" MFRIT coefficients are obviously orthogonal within each block, and also with other blocks since they do not overlap. Furthermore, these basis functions are orthogonal with the constant function on their block, and thus orthogonality holds across levels as well. Consequently, the MFRIT is an orthonormal transform.

By collecting the MFRIT coefficients into groups depending on their scales and directions, we obtain a subband-like decomposition with J scales, where level i has p_i directions. When $p_i = 2$, the orthonormal FRIT using the Haar DWT is the same as the 2×2 Haar DWT (see Figure 5.9). Therefore the MFRIT scheme includes the multilevel 2-D Haar DWT. In general, when $p_i > 2$, the MFRIT offers more directions than the 2-D DWT.

5.6 Numerical Experiments

5.6.1 Non-linear Approximation

Following the study of the efficiency of the ridgelet transform in the continuous domain on the truncated Gaussian functions [21], we first perform numerical comparison on a 256×256 image of the function: $f(x_1, x_2) = 1_{\{x_2 < 2x_1 + 0.5\}} e^{-x_1^2 - x_2^2}$, using four 2-D transforms: DCT, DWT, FRAT and FRIT. For the FRAT and FRIT, we extend the image size to the next prime number, 257, by replicating the last pixel in each row and column. The wavelet used in both the DWT and FRIT is the "least asymmetric" orthogonal wavelet with 8-taps filters [42].

Our initial experiments indicate that in order to achieve good results, it is necessary to apply the DCT instead of the DWT to capture some strong periodic FRAT projections due to the "wrap around" effect (we refer to the remarks at the end of Section 5.4). Without resorting to adaptive methods, we employ a simple, pre-defined scheme where the least "wrap around" FRAT projections – the ones with $\|(a_k^\star, b_k^\star)\| \leq D$ – use DWT, while all the others use DCT. We set $D = 3$ in our experiments, which means in the FRIT, only 16 FRAT projections are represented by the DWT.

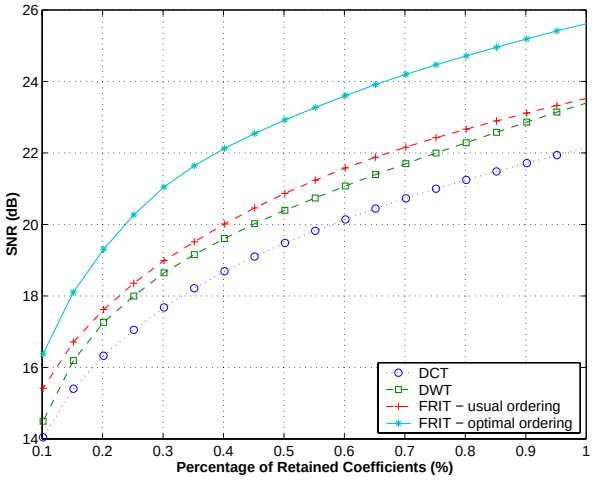
The comparison is evaluated in terms of the non-linear approximation power, i.e. the ability of reconstructing the original image, measured by signal-to-noise ratios (SNR's), using the N largest magnitude transform coefficients. Figure 5.11 shows the results. We omit the FRAT since its performance is much worse than the others. Clearly the FRIT achieves the best result, as expected from the continuous theory. Furthermore, the new ordering of the FRAT coefficients is crucial for the FRIT in obtaining good performance.

Our next test is an "object" image of size 256×256 with straight edges. Figure 5.12 shows the images obtained from non-linear approximation using the DWT and FRIT. As can be seen, the FRIT correctly picks up the edges using the first few significant coefficients and produces visually better approximated images.

To gain an insight into the FRIT, Figure 5.13(a) shows the top five FRAT projections for the "object" image that contain most of the energy, measured in the l_2-norm. Those projections correspond to the directions that have discontinuities across, plus the horizontal and vertical directions. Therefore, we see that at first the FRAT compacts most of the energy of the image into a few projections (see Figure 5.13(b)), where the linear discontinuities create "jumps". Next, taking the 1-D DWT on those projections, which are mainly smooth, compacts the energy further into a few FRIT coefficients.

5.6.2 Image Denoising

The motivation for the FRIT-based image denoising method is that in the FRIT domain, linear singularities of the image are represented by a few large coefficients, whereas randomly located noisy singularities are unlikely to produce



(a) (b)

Figure 5.11: (a) Test image: a truncated Gaussian image of size 256×256 that represents the function $f(x_1, x_2) = 1_{\{x_2 < 2x_1 + 0.5\}} e^{-x_1^2 - x_2^2}$. (b) Comparison of non-linear approximations using four different 2-D transforms: DCT, DWT, FRIT with usual ordering and FRIT with optimal ordering.

significant coefficients. By contrast, in the DWT domain, both image edges and noisy pixels produce similar amplitude coefficients. Therefore, a simple thresholding scheme for FRIT coefficients can be very effective in denoising images that are piecewise smooth away from singularities along straight edges.

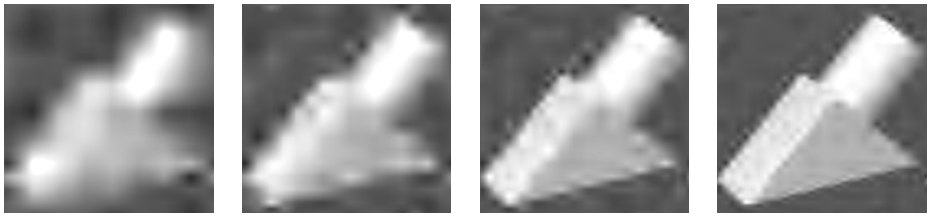
We consider a simple case where the original image is contaminated by an additive zero-mean Gaussian white noise of variance σ^2. With an orthogonal FRIT, the noise in the transform domain is also Gaussian white of the same variance. Therefore it is appropriate to apply the thresholding estimators that were proposed in [61] to the FRIT coefficients. More specifically, our denoising algorithm consists of the following steps:

Step 1: Applying FRIT to the noisy image.

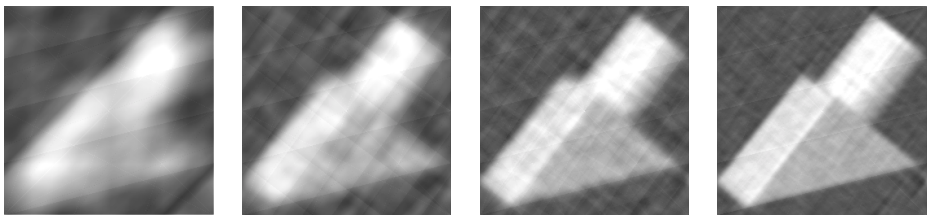
Step 2: Hard-thresholding of FRIT coefficients with the universal threshold $T = \sigma\sqrt{2logN}$ where $N = p^2$ pixels.

Step 3: Inverse FRIT of the thresholded coefficients.

For an image which is smooth away from linear singularities, edges are visually well restored after Step 3. However due to the periodic property of the FRIT, strong edges sometimes create "wrap around" effects which are visible in the smooth regions of the image. In order to overcome this problem, we optionally employ a 2-D adaptive filtering step.



(a) Using DWT



(b) Using FRIT

Figure 5.12: From left to right, reconstructed images from the 32, 64, 128 and 256 most significant coefficients of the DWT and FRIT, out of 65536 coefficients.

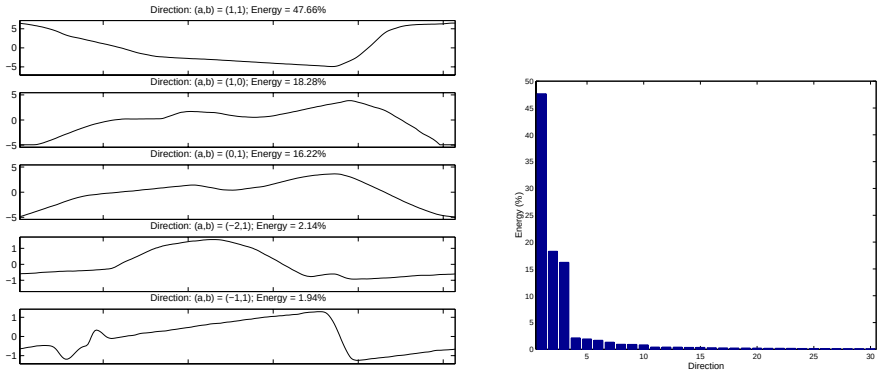
Step 4: (Optional) Adaptive Wiener filtering to reduce the "wrap around" effect. In some cases, this can enhances the visual appearance of the restored image.

The above FRIT denoising algorithm is compared with the wavelet hard-thresholding method using the same threshold value. Figure 5.14 shows the denoising results on the real image. The FRIT is clearly shown to be more effective than the DWT in recovering straight edges, as well as in term of SNR's.

5.7 Conclusion and Discussion

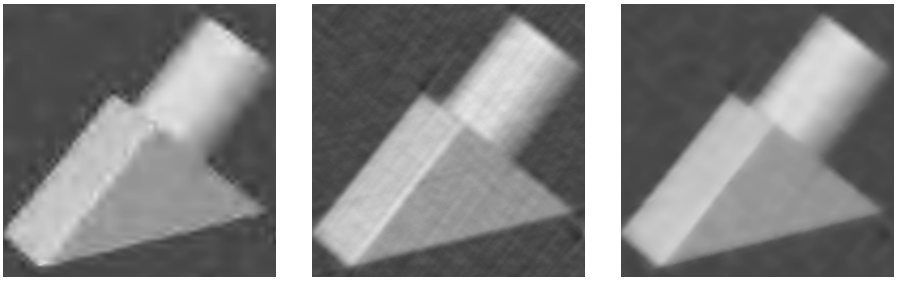
We presented a new family of discrete orthonormal transforms for images based on the ridgelet idea. Owning to orthonormality, the proposed ridgelet transform is self-inverting – the inverse transform uses the same algorithm as the forward transform – and has excellent numerical stability. Experimental results indicate that the FRIT offers an efficient representation for images that are smooth away from line discontinuities or straight edges.

However, it is important to emphasize that the ridgelet transform is only suited for discontinuities along straight lines. For complex images, where edges are mainly along bended curves and there are texture regions (which generate point discontinuities), the ridgelet transform is not optimal. Therefore, a more



(a) (b)

Figure 5.13: (a) Top five FRAT projections of the "object" image that contain most of the energy. (b) Distribution of total input image energy among FRAT projections. Only the top 30 projections are shown in the descending order.



(a) Using DWT; SNR = 19.78 dB.

(b) Using FRIT; SNR = 19.67 dB

(c) Using FRIT and Wiener filter; SNR = 21.07 dB.

Figure 5.14: Comparison of denoising on the "object" image.

practical scheme in employing the ridgelet transform is to first utilize a quadtree division of images into suitable blocks where edges look straight and then apply the discrete ridgelet transform to each block. Another scheme is to use the ridgelet transform as the building block in a more localized construction such as the *curvelet* transform [20].

Appendix 5.A Orthogonal Wavelet Transform for Non Dyadic Length Signals

In the construction of the orthonormal FRIT, we need wavelet bases for signals of prime length p. In addition, those bases have to satisfy the *Condition Z* in Lemma 5.1. Let $n = 2^J$ be the nearest dyadic number to p that is smaller than or equal to p. Suppose that $p - n$ is small, then one simple way of taking the wavelet transform on a sequence of p samples is to apply the usual wavelet transform on the first n samples and then extend it to cover the remaining $p-n$ samples.

Let $\{\boldsymbol{v}_m : m \in Z_n\}$ to be the basis vectors of an orthonormal wavelet transform of length n with J decomposition levels. We assume periodic extension is used to handle the boundary. Suppose that $\boldsymbol{v}_0$ corresponds to the single scaling coefficient or the mean value, then all other vectors must have zero mean (see Lemma 5.1). Denote $c^{\{k\}}$ be the vector with k entries, all equal to c. Consider the following p vectors defined in $\mathbb{R}^p$

$$\begin{aligned}
\boldsymbol{w}_0 &= (1^{\{p\}}) \,/\, s_0 \\
\boldsymbol{w}_1 &= (1^{\{p-1\}}, -p+1) \,/\, s_1 \\
\boldsymbol{w}_2 &= (1^{\{p-2\}}, -p+2, 0) \,/\, s_2 \\
&\cdots \\
\boldsymbol{w}_{p-n} &= (1^{\{p-n\}}, -p+n, 0^{\{n-1\}}) \,/\, s_{p-n} \\
\boldsymbol{w}_{p-n+1} &= (\boldsymbol{v}_1, 0^{\{p-n\}}) \\
&\cdots \\
\boldsymbol{w}_{p-1} &= (\boldsymbol{v}_{n-1}, 0^{\{p-n\}}).
\end{aligned}$$

Here s_k is the scale factor such that $\|\boldsymbol{w}_k\| = 1$. The orthogonality of the new set $\{\boldsymbol{w}_k : k \in Z_p\}$ can be easily verified given the fact that $\{\boldsymbol{v}_m : 1 \le m < n\}$ are orthonormal vectors with zero mean. Therefore, $\{\boldsymbol{w}_k : k \in Z_p\}$ is an orthonormal basis for $\mathbb{R}^p$ that satisfies the *Condition Z*. For a length p input vector $\boldsymbol{x} = (x_0, x_1, \ldots, x_{p-1})$, the transform coefficients correspond to $\boldsymbol{w}_k$, where $p - n \le k \le p - 1$, can be computed efficiently using the usual DWT with J levels on the first n samples $\boldsymbol{x}' = (x_0, x_1, \ldots, x_{n-1})$. The last scaling coefficient is then replaced by $p - n + 1$ coefficients corresponding to the basis vectors $\boldsymbol{w}_k,\ k = 0, \ldots, p - n$. Thus the new basis in $\mathbb{R}^p$ also has fast transforms.

Chapter 6

Texture Retrieval using Directional Multiresolution Representations

6.1 Introduction

6.1.1 Motivation

Digital image libraries are becoming more widely used as more visual information is put in digital form as well as on-line. To improve human access, however, there must be an effective and precise method for users to search, browse, and interact with these collections and to do so in a timely manner. As a result, content-based image retrieval (CBIR) from unannotated image databases has been a fast growing research area recently: see [155] for a recent extensive review on the subject.

We consider a simple architecture of a typical CBIR system (Figure 6.1), where there are two major tasks. The first one is *Feature Extraction* (FE), where a set of features, called image signatures, is generated to accurately represent the content of each image in the database. A signature is much smaller in size than the original image, typically on the order of hundreds of elements (rather than millions). The second task is *Similarity Measurement* (SM), where a distance between the query image and each image in the database using their signatures is computed so that the top N "closest" images can be retrieved. Typically, the features used in CBIR systems are low-level image features such as color, texture, shape and layout. In this work, we focus on the use of texture information for image retrieval.

Some of the most popular texture extraction methods for retrieval are based on filtering or wavelet-like approaches [95, 24, 156, 167, 107, 179, 135]. Essen-

[0]This chapter includes research conducted jointly with Martin Vetterli and Aurélie C. Lozano [60, 53, 50, 59].

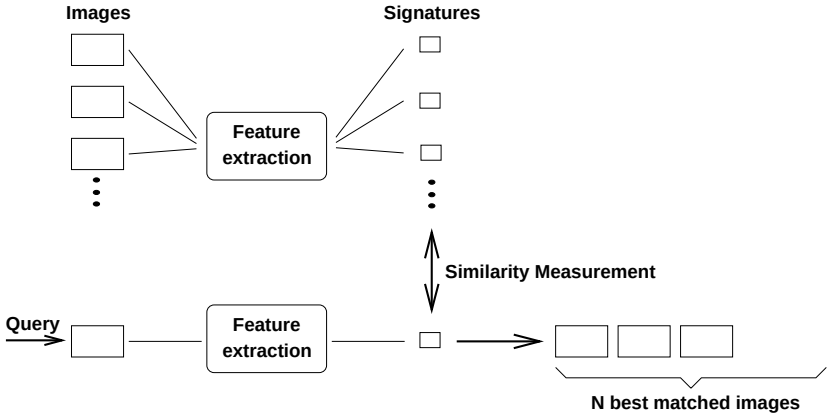


Figure 6.1: Image retrieval system architecture, where there are two major tasks: Feature Extraction and Similarity Measurement.

tially, these methods measure energy (possibly weighted) at the output of filter banks as extracted features for texture discrimination. The basic assumption of these approaches is that the energy distribution in the frequency domain identifies a texture. Besides providing acceptable retrieval performance from large texture databases, those approaches are partly supported by physiological studies of the visual cortex [82, 44]. Furthermore, as wavelets are a core technology in the next generation of still image coding format, JPEG-2000 [36], the choice of wavelet features enables the implementation of retrieval systems that can work directly in the compressed domain. Other possible transforms are wavelet packets, wavelet frames and Gabor wavelet transforms.

Finding good similarity measures between images based on some feature set is a challenging task. On the one hand, the ultimate goal is to define similarity functions that match with human perception, but how humans judge the similarity between images is a topic of ongoing research [158]. Perceptual studies [162, 136] identified texture dimensions by conducting experiments that asked observers to group textures according to perceived similarity. The detected perceptual criteria and rules for similarity judgment from this type of subjective experiments can be used in building image retrieval system [114]. On the other hand, many current retrieval systems take a simple approach by using typically norm-based distances (e.g. Euclidean distance) on the extracted feature set as a similarity function [155]. The main premise behind these CBIR systems is that given a "good set" of features extracted from the images in the database (the ones that significantly capture the content of images) then for two images to be "similar" their extracted features have to be "close" to each other. Therefore, any reasonable similarity functions defined on the features space should perform well. Sometimes, weighting factors are necessary to normalize extracted features over the entire database to comparable ranges so that they have approximately the same influence on the overall distance.

Note that this "global" normalization process is *different* with the one often

used in classification problems where the normalized factors are computed using a training set of feature vectors from each class. Furthermore, the commonly used inverse variance weighted Euclidean distance as in CBIR [69] is questionable in the case of a feature component that has small global variance, thus leading to a large weight in the overall distance. By contrast, it can be argued that a small variation component should have little discrimination power and should thus carry a small weight in the overall distance.

6.1.2 Our Approach and Related Works

In this work we consider jointly the problems of FE and SM in texture retrieval using a statistical approach. Our point is that, given only a low-level representation, statistical modeling provides a natural mean to formulate the retrieval problem, as is typically done in pattern recognition. Considering the two related retrieval tasks FE and SM as estimation and detection problems, respectively, provides us with a justified way of defining similarity functions on the feature space. The implication of this approach is twofold. First, it provides a confidence on the optimality of the defined similarity function under some explicit assumptions. Secondly, as we will see, this approach provides a common ground for many existing similarity functions by simply modifying the underlying assumptions.

Statistical modeling has been used in CBIR systems before. Perhaps the most well-known examples are the use of histograms to capture the distribution of image features such as color [161]. Wouwer et al. [179] employed generalized Gaussian density functions to represent texture images in the wavelet domain. The model parameters are estimated using a method of moment matching, and the similarity function is again defined as weighted Euclidean distances on extracted model parameters. Independently of our work, Vasconcelos and Lippman [171] recently took a similar approach where they introduced a probabilistic formulation of the CBIR problem as a common ground for several currently used similarity functions.

As an important case of CBIR, we demonstrate in this work the application of the statistical framework in the wavelet-based texture retrieval problem. The statistical approach fits nicely into this case, since a texture image is often regarded as a realization of an underlying stochastic process. In the end, we will briefly discuss how such approach can be applied to other features and integrated into more general image retrieval systems.

The outline of this chapter is as follows. In the next section, we set up the CBIR problem in a general statistical framework. In Section 6.3, we apply this framework to the wavelet-based texture retrieval application where wavelet coefficients in each subband are independently modeled by a generalized Gaussian distribution (GGD). Section 6.4 presents experimental results on a large texture image database, which indicate a significant improvement in the retrieval rate using the new approach. While having low complexity, the independent distribution model ignore the dependency of wavelet descriptors across scales

and orientations. Section 6.5 presents a coherent texture model based on the wavelet-domain hidden Markov models (HMM's) [38] that captures both wavelet subband marginal distributions and inter-subband dependencies, while being rotation invariant. Experiments with the wavelet-domain HMM's are given in Section 6.6. Finally, Section 6.7 concludes with some discussions.

6.2 Content-based Image Retrieval in a Statistical Framework

6.2.1 General Setting

The problem of searching for the top N images similar to a given query image from a database of total M images ($N \ll M$) can be formulated as a multiple hypotheses problem.[1] The query image $\mathcal{I}_q$ is represented by its data set $\boldsymbol{x} = (x_1, x_2, \ldots, x_L)$, which is typically obtained after a pre-processing stage. Each candidate image in the database $\mathcal{I}_i : i = 1, 2, \ldots, M$ is assigned with a hypothesis $\mathcal{H}_i$. The goal is to select among the M possible hypotheses the N best ones (with a ranking order) that describe the data $\boldsymbol{x}$ from the query image.

To select the N top matches from those M hypotheses we can use the multiple hypotheses testing argument recursively. That is, we first choose the best one among the M possible hypotheses $\{\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_M\}$, and then we choose the next best one among the remain $(M-1)$ hypotheses, and keep doing so for N times. Under the common assumption that all prior probabilities of the hypotheses are equal, it can be shown [130] that, for each recursive step the optimum rule (with the minimum probability of error criterion) is to choose the hypothesis with the highest likelihood among the possible ones. Thus for CBIR, it is optimal to select N hypotheses with highest likelihood, i.e. $\mathcal{H}_{k_1}, \mathcal{H}_{k_2}, \ldots, \mathcal{H}_{k_N}$ where

$$p(\boldsymbol{x}|\mathcal{H}_{k_1}) \geq p(\boldsymbol{x}|\mathcal{H}_{k_2}) \geq \ldots \geq p(\boldsymbol{x}|\mathcal{H}_{k_N}) \geq p(\boldsymbol{x}|\mathcal{H}_i) \quad i \neq k_j \ (j = 1, 2, \ldots, N). \quad (6.1)$$

This is referred to as the *maximum likelihood* (ML) selection rule. The problem with (6.1) is that it requires M computational steps with a typically large data set $\boldsymbol{x}$. This turns out to be impractical in CBIR applications since this operation has to be done *on-line* in the interactive mode. Therefore, we need to find an approximation with much less computational cost.

In the *parametric* approach, the conditional probability density $p(X|\mathcal{H}_i)$ is modeled by a member of a family of probability density functions (PDF's), denoted by $p(X; \boldsymbol{\theta}_i)$ where $\boldsymbol{\theta}_i$ is a set of model parameters. With this setting, the extracted features for the image $\mathcal{I}_i$ is the estimated model parameter $\hat{\boldsymbol{\theta}}_i$,

[1] However the term "hypotheses" is used here in a loose sense. Since in CBIR applications, the search is not for the *exact match* but rather for *most similar* ones, hence we can allow for more than one hypothesis to be valid.

which is computed in the FE step. We denote the space of model parameters as $\boldsymbol{\Theta}$.

Consider the query data $\boldsymbol{x} = (x_1, x_2, \ldots, x_L)$ as an independent and identically distributed (i.i.d.) sequence from the model $p(X; \boldsymbol{\theta}_q)$ of the query image. Then for large L, using the weak law of large number, the ML selection rule (6.1) is equivalent to maximizing

$$\begin{aligned} \frac{1}{L} \log p(\boldsymbol{x}; \boldsymbol{\theta}_i) &= \frac{1}{L} \sum_{j=1}^{L} \log p(x_j; \boldsymbol{\theta}_i) \\ &\stackrel{L\to\infty}{\longrightarrow} E_{\boldsymbol{\theta}_q} [\log p(X; \boldsymbol{\theta}_i)] \quad \text{in probability (i.p.)} \\ &= \int p(x; \boldsymbol{\theta}_q) \log p(x; \boldsymbol{\theta}_i) dx. \end{aligned}$$

This can be seen as equivalent to minimizing the *Kullback-Leibler distance* (KLD) or the *relative entropy* [37] between the two PDF's $p(x; \boldsymbol{\theta}_q)$ and $p(x; \boldsymbol{\theta}_i)$

$$D(p(X; \boldsymbol{\theta}_q) || p(X; \boldsymbol{\theta}_i)) = \int p(x; \boldsymbol{\theta}_q) \log \frac{p(x; \boldsymbol{\theta}_q)}{p(x; \boldsymbol{\theta}_i)} dx \tag{6.2}$$

Under the same asymptotic condition (L is large), if the FE step uses a *consistent* estimator, which ensures the estimated parameter $\hat{\boldsymbol{\theta}}$ converges to the true parameter $\boldsymbol{\theta}$, then the distance (6.2) can be computed using the *estimated* model parameters $\hat{\boldsymbol{\theta}}_q$ and $\hat{\boldsymbol{\theta}}_i$. For such consistent estimator, we could employ the ML estimator [88], which means that for the query image, it computes

$$\hat{\boldsymbol{\theta}}_q = \arg \max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \log p(\boldsymbol{x}; \boldsymbol{\theta}). \tag{6.3}$$

In summary, by combining FE and SM into a joint modeling and classification scheme, the optimum ML selection rule can be *asymptotically* realized (as the data sets for each image become large) by:

Feature Extraction: Given the data from each image, extracting features as estimated model parameters using a consistent estimator such as the ML estimator.

Similarity Measurement: To select the top N matches to a query, the images in the database are ranked based on the KLD's between the estimated model for the query and estimated models for each image.

The advantage of this scheme is that the SM step can be computed entirely on the estimated model parameters, which are typically small size, so that it can meet the timing constraint of the CBIR application. The method is generic as it allows the use of any feature data and statistical models for indexed images. Such image models can incorporate the knowledge from perceptual studies to closely match human judgment.

We point out that the Kullback-Leibler distance has been used in comparing images (e.g. in [26, 16]) but, to our knowledge, its use has not yet been

sufficiently justified in the context of the image retrieval problem by jointly considering the two related tasks FE and SM.

To combine the KLD's from multiple data sets, such as from different channels or feature sets, we can use the *chain rule* [37] which states that the KLD between two joint PDF's $p(X,Y)$ and $q(X,Y)$ is

$$D(p(X,Y)||q(X,Y)) = D(p(X)||q(X)) + D(p(Y|X)||q(Y|X)). \tag{6.4}$$

Especially, when data is considered to be independent between combined sets then the joint KLD is simply the sum of KLD's from each set. Finally, the convexity property of KLD permits it to be used in an efficient search scheme using multiscale representations [26].

6.2.2 Relation to Histogram Methods

Histograms have been used since the early days of image retrieval, especially for representing color features [161], as well as for texture or local geometric properties [155]. In this section, we demonstrate that the histogram method can be interpreted through our statistical approach by using an appropriate model setup.

Let us partition the range of image data into R disjoint intervals of equal length, $\{S_1, S_2, \ldots, S_R\}$. Now consider the family of piecewise constant densities, defined as

$$p(x;\theta) = p_i \quad \text{for } x \in S_i, \quad i = 1, \ldots, R. \tag{6.5}$$

Here the set of model parameters is $\boldsymbol{\theta} := (p_1, \ldots, p_R)$, where $p_i \geq 0$ $(i = 1, \ldots, R)$ and $\sum_{i=1}^{R} p_i = 1$.

Given a sequence of i.i.d. data samples from an image, $\boldsymbol{x} = (x_1, \ldots, x_L)$, using the ML estimator in our statistical retrieval the FE step amounts to computing the feature vector $\hat{\boldsymbol{\theta}}$ where,

$$\begin{aligned} \hat{\boldsymbol{\theta}} &= \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \sum_{i=1}^{L} \log p(x_i;\boldsymbol{\theta}) \\ &= \arg\max_{\sum_{k=1}^{R} p_k = 1,\, p_k \geq 0} \sum_{k=1}^{R} n_k \log p_k. \end{aligned} \tag{6.6}$$

Here we denote n_k the number of data samples in $(x_1, \ldots, x_L)$ that belong to the region S_k. Solving (6.6) using the Lagrange multiplier gives,

$$\hat{\boldsymbol{\theta}} := \boldsymbol{p} \quad \text{with } p_k = n_k/L, \quad k = 1, \ldots, R. \tag{6.7}$$

So the extracted feature vector $\boldsymbol{p}$ is in fact the normalized histogram of the image data $\boldsymbol{x}$. When searching for similar images given the query data $\boldsymbol{x}^{(q)}$ and its features $\boldsymbol{p}^{(q)}$, the following (discrete) KLD between the query image and

each candidate image $\mathcal{I}_i$ can be used to rank the images in the database

$$\begin{aligned} D(\boldsymbol{p}^{(q)}||\boldsymbol{p}^{(i)}) &= \sum_{k=1}^{R} p_k^{(q)} \log \frac{p_k^{(q)}}{p_k^{(i)}} \\ &= \sum_{k=1}^{R} p_k^{(q)} \log p_k^{(q)} - \sum_{k=1}^{R} p_k^{(q)} \log p_k^{(i)}. \end{aligned} \tag{6.8}$$

Whereas the ML selection rule is based on the following log-likelihoods

$$\begin{aligned} \log p(\boldsymbol{x}^{(q)}|\mathcal{H}_i) &= \sum_{j=1}^{L} \log[p(x_j^{(q)}; \boldsymbol{p}^{(i)})] \\ &= \sum_{k=1}^{R} n_k^{(q)} \log p_k^{(i)} \\ &= L^{(q)} \sum_{k=1}^{R} p_k^{(q)} \log p_k^{(i)}. \end{aligned} \tag{6.9}$$

Comparing (6.8) with (6.9) indicates that in this case, the ranking based on the KLD is *exactly* (rather than asymptotically) the same with the optimum ML selection rule.

A drawback of the histogram method is that it requires a large number of extracted features, typical several hundreds histogram bins, to capture accurately image information. Thus it leads to impractical complexity in both storage of image indices and retrieval timing. In the next section, we employ the wavelet transform and the generalized Gaussian density to efficiently solve the texture retrieval problem within our statistical framework.

6.3 Wavelet Coefficients Modeled using Generalized Gaussian Density

6.3.1 Wavelet Representation

Statistical modeling is much easier if some preprocessing is carried out on the input images. Typical preprocessing is done via transformation of image pixel values into a suitable space where simple models with a small number of parameters can describe the data. Wavelets have recently emerged as an effective tool to analyze texture information as they provide a natural partition of the image spectrum into multiscale and oriented subbands via efficient transforms [95, 24, 156, 167, 107, 179, 135]. Furthermore, since wavelets are used in major future image compression standards [36] and are also shown to be prominent in searching for images based on color and shape [83, 48], a wavelet-based texture retrieval system can be used effectively in conjunction with a compression system and retrieval systems using other image features (see Figure 6.2).

Using the assumption that the energy distribution in frequency domain identifies texture, traditional approaches computed energies of wavelet subband as

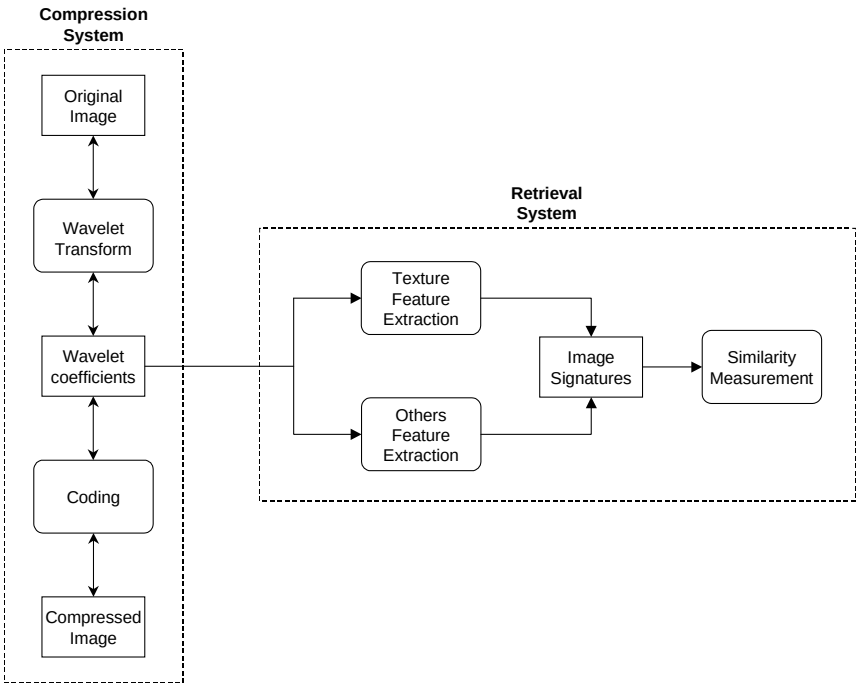


Figure 6.2: Block diagram of the wavelet-based texture retrieval system in conjunction with a compression system. Feature extraction tasks could be done concurrently and efficiently either when an image is compressed and inserted into the database or when an input compressed image is decoded.

texture features. Commonly, $\mathbf{L}^1$ and $\mathbf{L}^2$ norms are used as measures.[2] More specifically, given the wavelet coefficients $x_{i,1}, x_{i,2}, \ldots, x_{i,L}$ at the i-th subband, typically the following two values are used as features:

$$f_i^{(1)} = \frac{1}{L}\sum_{j=1}^{L}|x_{i,j}|, \text{ and} \tag{6.10}$$

$$f_i^{(2)} = \left(\frac{1}{L}\sum_{j=1}^{L}x_{i,j}^2\right)^{1/2}. \tag{6.11}$$

On the other hand, statistical approaches treat texture analysis as a probability inference problem (e.g. see [181]). A natural extension of the energy method is to model a texture by the marginal densities of wavelet subband coefficients. This is justified by recent psychological research on human texture perception which suggests that two homogeneous textures are often difficult to discriminate if they produce similar marginal distributions of responses from

[2]This is an abuse of terminology since strictly speaking $\mathbf{L}^1$ norm is not an energy function. Sometimes it is chosen due to its simplicity. Results from several studies indicate no general conclusion in favor of a particular measure.

a bank of filters [11]. In fact, Heeger and Bergen [77] successfully synthesized many natural looking texture images by matching the histograms of filter responses from a wavelet-liked transform. More accurate texture models can be obtained via a *fusion* of marginal distributions using minimax entropy principles [181] or by taking into account the joint statistics of wavelet coefficients across subbands [153]. However, considering complexity as a major constraint in the image retrieval application, in this work we simply characterize texture images via marginal distributions of their wavelet subband coefficients. Still, this representation of texture is more precise than the ones that use wavelet subband energies alone. But more importantly, the statistical modeling leads to a more justifiable way of defining similarity functions between images.

6.3.2 Generalized Gaussian Density Modeling of Wavelet Coefficients

Experiments show that a good PDF approximation for the marginal density of coefficients at a particular subband produced by various type of wavelet transforms may be achieved by adaptively varying two parameters of the *generalized Gaussian* density (GGD) [102, 148, 179, 115], which is defined as:

$$p(x;\alpha,\beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(|x|/\alpha)^{\beta}}, \tag{6.12}$$

where $\Gamma(.)$ is the Gamma function, i.e. $\Gamma(z) = \int_0^{\infty} e^{-t} t^{z-1} dt,\ z > 0$.

Here α models the width of the PDF peak (standard deviation), while β is inversely proportional to the decreasing rate of the peak. Sometimes, α is referred to as the *scale* parameter while β is called the *shape* parameter. The GGD model contains the Gaussian and Laplacian PDF's as special cases, using $\beta = 2$ and $\beta = 1$, respectively.

Within a CBIR statistical framework, the desired estimator in our case is the maximum-likelihood (ML) estimator. Furthermore, in [170] evaluation of accuracy of estimates for both large and small samples for GGD models among classic statistical methods shows that the ML estimator is significantly superior for heavy-tailed distribution (which is often the case for subband coefficients). We now describe the ML estimator for GGD.

Let us define the likelihood function of the sample $\boldsymbol{x} = (x_1, \ldots, x_L)$ having independent component as

$$L(\boldsymbol{x};\alpha,\beta) = \log \prod_{i=1}^{L} p(x_i;\alpha,\beta),$$

where α and β are parameters to be estimated. It was shown in [170] that in this case the following likelihood equations have a unique root in probability,

which is indeed the maximum-likelihood estimator

$$\frac{\partial L(\boldsymbol{x};\alpha,\beta)}{\partial \alpha} = -\frac{L}{\alpha} + \sum_{i=1}^{L} \frac{\beta\ |x_i|^{\beta}\,\alpha^{-\beta}}{\alpha} = 0 \tag{6.13}$$

$$\frac{\partial L(\boldsymbol{x};\alpha,\beta)}{\partial \beta} = \frac{L}{\beta} + \frac{L\Psi(1/\beta)}{\beta^2} - \sum_{i=1}^{L} \left(\frac{|x_i|}{\alpha}\right)^{\beta} \log\left(\frac{|x_i|}{\alpha}\right) = 0, \tag{6.14}$$

here $\Psi(.)$ is the digamma function [1], i.e. $\Psi(z) = \Gamma'(z)/\Gamma(z)$.

Fix $\beta > 0$ then (6.13) has a unique, real, and positive solution as:

$$\hat{\alpha} = \left(\frac{\beta}{L}\sum_{i=1}^{L} |x_i|^{\beta}\right)^{1/\beta}. \tag{6.15}$$

Substitute this into (6.14), the shape parameter β is the solution of the following *transcendental* equation:

$$1 + \frac{\Psi(1/\hat{\beta})}{\hat{\beta}} - \frac{\sum_{i=1}^{L} |x_i|^{\hat{\beta}} \log|x_i|}{\sum |x_i|^{\hat{\beta}}} + \frac{\log\left(\frac{\hat{\beta}}{L}\sum_{i=1}^{L} |x_i|^{\hat{\beta}}\right)}{\hat{\beta}} = 0, \tag{6.16}$$

which can be solved numerically. We propose an effective determination of $\hat{\beta}$ using the Newton-Raphson iterative procedure [88] with the initial guess from the moment method described in [148]. This algorithm is detailed in Appendix 6.A. Experiments show that typically only around three iteration steps are required to obtain solutions with an accuracy of the order of 10^{-6}.

Figure 6.3 shows a typical example of a histogram of wavelet subband coefficients together with a plot of the fitted GGD using the ML estimator. The fits are generally quite good. As a result, with only two parameters for the GGD, we can accurately capture the marginal distribution of wavelet coefficients in a subband that otherwise would require hundreds of parameters by using histogram. This significantly reduces the storage of the image features, as well as the computational complexity in similarity measurement.

6.3.3 Similarity Measurement between GGD's

Given the GGD model, the PDF of wavelet coefficients in each subband can be completely defined via two parameters α and β. Substitute (6.12) into (6.2) and after some manipulations we obtain the following closed form for the Kullback-Leibler distance (KLD) between two GGD's as:

$$D(p(.;\alpha_1,\beta_1)||p(.;\alpha_2,\beta_2)) = \log\left(\frac{\beta_1\alpha_2\Gamma(1/\beta_2)}{\beta_2\alpha_1\Gamma(1/\beta_1)}\right) + \left(\frac{\alpha_1}{\alpha_2}\right)^{\beta_2} \frac{\Gamma((\beta_2+1)/\beta_1)}{\Gamma(1/\beta_1)} - \frac{1}{\beta_1}. \tag{6.17}$$

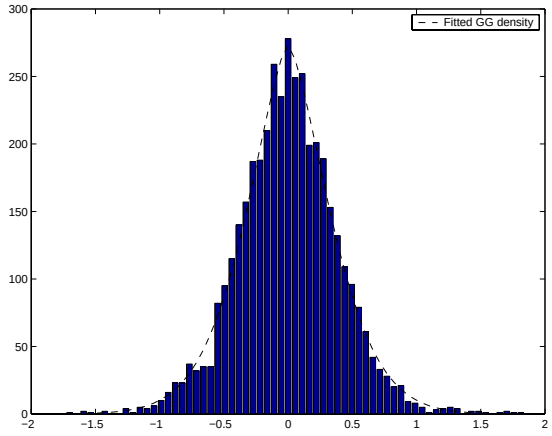


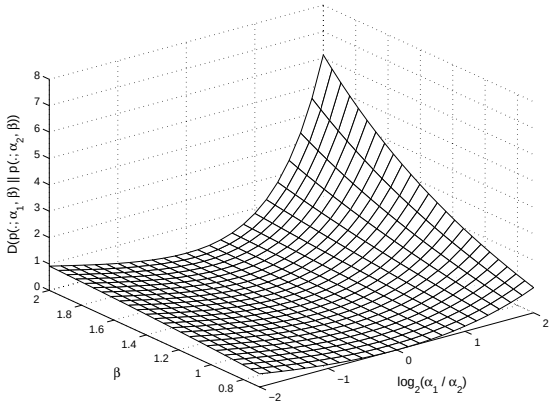
Figure 6.3: Wavelet subband coefficient histogram fitted with a generalized Gaussian density. Example for the Fabric15 subimage of size 128 $\times$ 128 at the highest horizontal frequency scale. The estimated parameters are: $\alpha = 0.46$ and $\beta = 1.52$.

Therefore, the similarity measurement between two wavelet subbands can be computed very effectively using the model parameters. Furthermore, applying (6.4) with the reasonable assumption that wavelet coefficients in different subbands are independent, the overall similarity distance between two images is precisely the sum of KLD's given in (6.17) between corresponding pairs of subbands. That is, if we denote $\alpha_i^{(j)}$ and $\beta_i^{(j)}$ as the extracted texture features from the wavelet subband j of the image $\mathcal{I}_i$ then the overall distance between two images $\mathcal{I}_1$ and $\mathcal{I}_2$ (where $\mathcal{I}_1$ is the query image) is the sum of all the distances across all wavelet subbands:

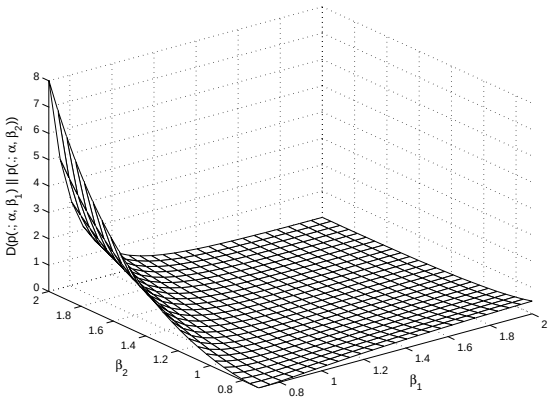
$$D(\mathcal{I}_1, \mathcal{I}_2) = \sum_{j=1}^{B} D(p(.; \alpha_1^{(j)}, \beta_1^{(j)}) || p(.; \alpha_2^{(j)}, \beta_2^{(j)})), \tag{6.18}$$

where B is the number of analyzed subbands. Thus the KLD theory provides us with a justified way of combining distances into an overall similarity measurement, and no normalization on the extracted features is needed.

The distance function defined in (6.17) is a function of three variables: the ratio of two scales α_1/α_2 and two shape parameters β_1 and β_2. Figure 6.4 plots the distance function when the two distributions have the same shape parameter and when they have the same scale parameter. The chosen ranges for the ratio α_1/α_2 as $[0.25, 4]$ comes from the fact that we are only interested in the two relatively close distributions when searching for most similar images. The selected range for β as $[0.7, 2.0]$ is based on the experimental results (refer to Section 6.4.3)



(a) When $\beta_1 = \beta_2 = \beta$.



(b) When $\alpha_1 = \alpha_2 = \alpha$.

Figure 6.4: Kullback-Leibler distance between two generalized Gaussian densities given in (6.17).

6.3.4 A Special Case of GGD

To gain more insight on the similarity function (6.17), let us consider the special case of a GGD where the shape parameter β is fixed. That means we model the wavelet coefficients using the following single parameter α distribution family.

$$\left\{ p_\beta(.;\alpha), \alpha \in \mathbb{R}_+ \ : \ p_\beta(x;\alpha) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(|x|/\alpha)^\beta} \right\}. \tag{6.19}$$

This simplified model is often used in practical image processing problems like denoising [115]. From the sample data sequence $\boldsymbol{x} = \{x_1, x_2, \ldots, x_L\}$, the

extracted feature is just the estimated parameter α as given in (6.15)

$$\hat{\alpha}_{\boldsymbol{x}} = \left(\frac{\beta}{L} \sum_{i=1}^{L} |x_i|^{\beta} \right)^{1/\beta}. \tag{6.20}$$

The KLD between two PDF's from the family given in (6.19) is:

$$D(p_\beta(.;\alpha_1)||p_\beta(.;\alpha_2)) = \log\left(\frac{\alpha_2}{\alpha_1}\right) + \left(\frac{\alpha_1}{\alpha_2}\right)^{\beta} \frac{1}{\beta} - \frac{1}{\beta}. \tag{6.21}$$

Next, consider again the optimum ML selection rule. At a particular wavelet subband, denote $\boldsymbol{x} = \{x_1, x_2, \ldots, x_L\}$ the wavelet coefficients of the query image and $p_\beta(x;\alpha_k),\ k = 1, 2, \ldots, M$ as the estimated marginal distributions for each candidate images in the database. The ranking procedure should be based on the following normalized log-likelihood values, which can be simplified as

$$\begin{aligned} l_k(\boldsymbol{x}) &= \frac{1}{L}\sum_{i=1}^{L} \log p_\beta(x_i;\alpha_k) \\ &= \log\beta - \log 2 - \log\alpha_k - \log\Gamma(1/\beta) - \frac{\sum_{i=1}^{L}|x_i|^{\beta}}{L\alpha_k^{\beta}} \\ &= -\log\alpha_k - \left(\frac{\hat{\alpha}_{\boldsymbol{x}}}{\alpha_k}\right)^{\beta}\frac{1}{\beta} + C_\beta, \end{aligned}$$

where $\hat{\alpha}_{\boldsymbol{x}}$ is the extracted feature from the query data given in (6.20) and C_β is a constant only dependent on β.

The KLD's between the query model parameter and a candidate image model parameter is

$$\begin{aligned} d_k(\boldsymbol{x}) &= D\left(p_\beta(.|\hat{\alpha}_{\boldsymbol{x}})||p_\beta(.|\alpha_k)\right) \\ &= \log\alpha_k + \left(\frac{\hat{\alpha}_{\boldsymbol{x}}}{\alpha_k}\right)^{\beta}\frac{1}{\beta} - \log\hat{\alpha}_{\boldsymbol{x}} - \frac{1}{\beta}. \end{aligned}$$

So we can see that, with a given query data $\boldsymbol{x}$, in searching for the best matches, maximizing the log-likelihood $l_k(\boldsymbol{x})$ for $k = 1, 2, \ldots, M$ is *exactly* the same as minimizing the KLD's $d_k(\boldsymbol{x})$ for $k = 1, 2, \ldots, M$. Thus in this simplified case, the retrieval process using the KLD's provides the same result as the optimum ML selection rule that uses direct data from the query image. Note that in the general case, this is only true asymptotically when the size of data $L \to \infty$. Here, however, it is true *for every* L.

6.3.5 Relation to Energy-based Methods in the Laplacian Case

Furthermore, consider the case when the parameter β is fixed and equal 1. That is we are modeling the wavelet coefficients using the Laplacian distribution. The extracted feature from wavelet coefficients $\boldsymbol{x}$ of a particular subband is:

$$\hat{\alpha}_{\boldsymbol{x}} = \frac{\sum_{i=1}^{L}|x_i|}{L}. \tag{6.22}$$

This is precisely the $\mathbf{L}^1$-norm feature of wavelet coefficients as in (6.10).
From (6.21), the KLD between two Laplacian distribution is

$$D(p_1(.;\alpha_1)||p_1(.;\alpha_2)) = \log\left(\frac{\alpha_2}{\alpha_1}\right) + \frac{\alpha_1}{\alpha_2} - 1.$$

This is a convex function of α_2/α_1 and is minimum when $\alpha_2/\alpha_1 = 1$. Therefore in term of selecting the most similar images, we are only interested in the situation when the ratio α_2/α_1 is in the vicinity of 1. Using first-order Taylor approximation of $\log x$ around 1, $\log x \approx x - 1$ when $x \approx 1$, we have

$$\begin{aligned} D(p_1(.;\alpha_1)||p_1(.;\alpha_2)) &\approx \frac{\alpha_2}{\alpha_1} - 1 + \frac{\alpha_1}{\alpha_2} - 1 \\ &= \frac{(\alpha_2 - \alpha_1)^2}{\alpha_1\alpha_2}. \end{aligned}$$

Substitute this into (6.18), the overall similarity measurement between two images $\mathcal{I}_1$ and $\mathcal{I}_2$ becomes

$$D(\mathcal{I}_1, \mathcal{I}_2) \approx \sum_{j=1}^{B} \frac{(\alpha_2^{(j)} - \alpha_1^{(j)})^2}{\alpha_2^{(j)}\alpha_1^{(j)}}. \tag{6.23}$$

This distance is essentially the same as the popular weighted Euclidean distance between extracted features $\alpha_i^{(j)}$ where "global" normalization factors $w^{(j)} = \text{var}\{\alpha_i^{(j)} : i = 1, 2, \ldots, M\}$ are replaced by "local" normalization factors $w_{1,2}^{(j)} = \alpha_2^{(j)}\alpha_1^{(j)}$.

Therefore, we demonstrated that our statistical method with a GGD model on the wavelet coefficients can be particularized to closely resemble and thus provide a justification for the weighted Euclidean distance between $\mathbf{L}^1$-norms of wavelet subbands. This is an interesting fact since the two approaches are based on totally different assumptions. The former relies on an underlying stochastic process of the texture image while the later is based on the energy distribution in the frequency domain.

6.3.6 Summary of Different Forms of KLD

Let us summarize the different forms of KLD that we have seen so far. In Section 6.3.3 we introduced the general formula (6.17) for the KLD between GGD's for two wavelet subbands together with the overall similarity measurement between two images (6.18) as the sum of all the distances across wavelet subbands. Section 6.3.4 and 6.3.5 considered special cases to gain more insight of the technique as well as providing explanation for existing methods. For practical applications, as well as for the following experiments, the general form of KLD's in Section 6.3.3 are used.

6.4 Experiments with the Wavelet GGD Model

We used 40 textures obtained from the MIT Vision Texture (VisTex) database [113] and displayed them in Figure 6.5. These are real world 512×512 images

from different natural scenes. Only gray-scale levels of the images (computed from the luminance component) were used in the experiments. Since we define similar textures as subimages from a single original one, we selected texture images whose visual properties do not change too much over the image.

Each of the 512×512 images was divided into sixteen 128×128 non-overlapping subimages, thus creating a test database of 640 texture images. Furthermore, to eliminate the effect of common range in the gray level of subimages from a same original image and to make the retrieval task less biased, each subimage was individually normalized to zero mean and unit variance before the processing.

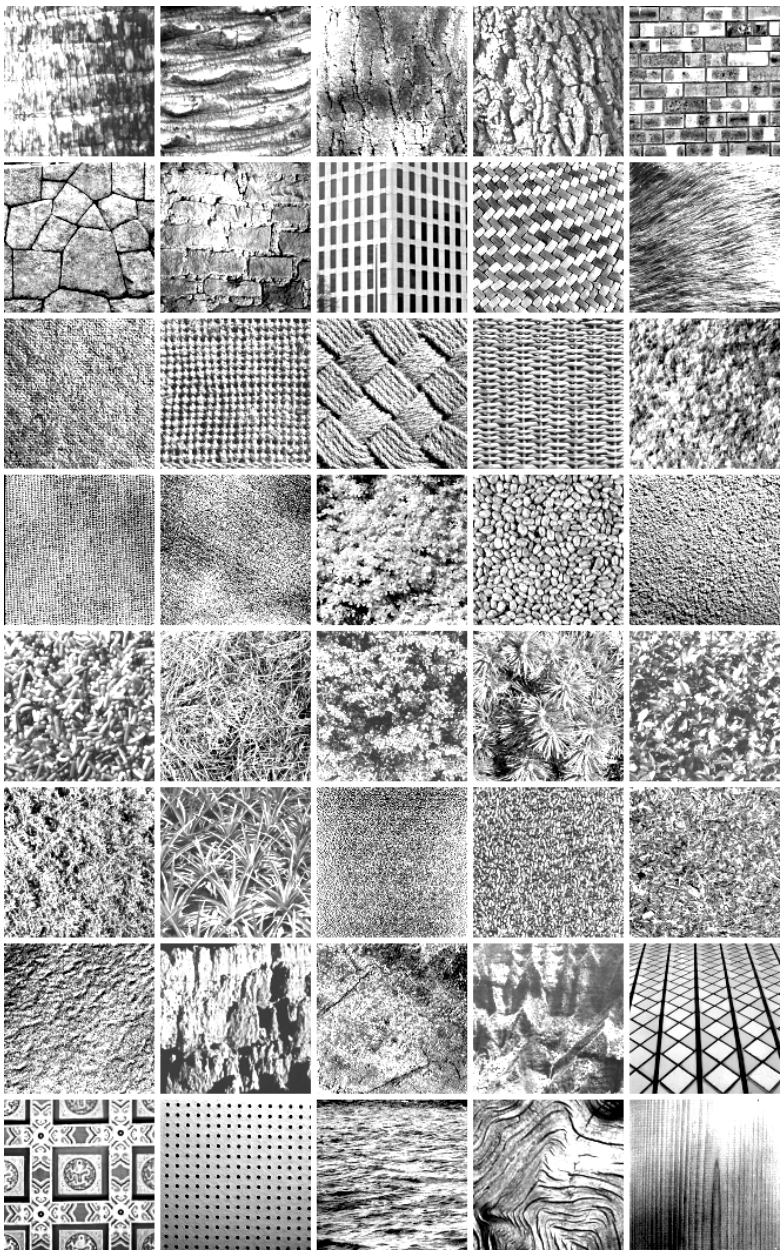


Figure 6.5: Texture images from the VisTex collection that are used in the experiments; from left to right and top to bottom: Bark0, Bark6, Bark8, Bark9, Brick1, Brick4, Brick5, Buildings9, Fabric0, Fabric4, Fabric7, Fabric9, Fabric11, Fabric14, Fabric15, Fabric17, Fabric18, Flowers5, Food0, Food5, Food8, Grass1, Leaves8, Leaves10, Leaves11, Leaves12, Leaves16, Metal0, Metal2, Misc2, Sand0, Stone1, Stone4, Terrain10, Tile1, Tile4, Tile7, Water5, Wood1, Wood2.

6.4.1 Testing the Texture Discrimination Power of the GGD Model

The accuracy of GGD's in modeling wavelet coefficients from texture images has been shown in [179] by fitting the estimated PDF curve with the actual histogram of the coefficients. In this section we explore another way of testing the GGD models in terms of comparing synthesized textures visually. This allow us to *visualize* the capability of the GGD models not only in capturing texture information but also in discriminating textures.

In this experiment, we employed the conventional pyramid wavelet decomposition with three levels using the Daubechies' maximally flat orthogonal filters of length 8 (D_4 filters) [42]. From a single image in the database (of size 128 $\times$ 128), two GGD parameters were estimated from each of 9 wavelet subbands (except for the lowest band or scale coefficients which corresponds to the approximation of the image) using the ML estimator described in the previous section. Our hypothesis is that those 18 (2×9) model parameters capture important texture-specific features and have discrimination power among texture classes.

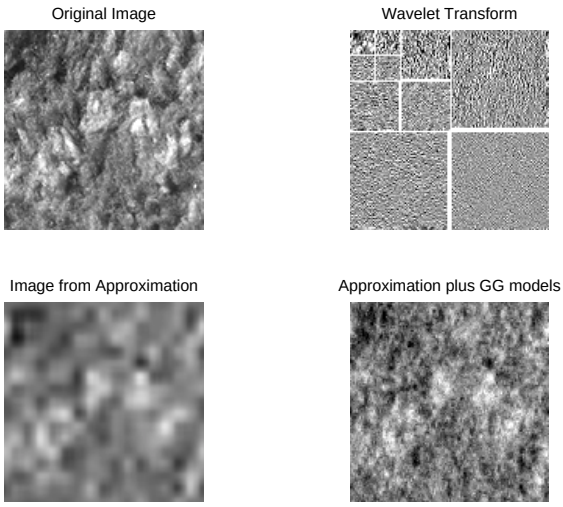
To show that, wavelet coefficients at each wavelet subband are generated as i.i.d. sample sequence from the GGD with parameters estimated from the true coefficients. Using those generated wavelet coefficients and the scale coefficients from the original image, the synthesis texture image is obtained by the inverse wavelet transform. Figure 6.6 shows the example results for two images of size 128×128 from different texture classes in our database. The comparison is also made with the reconstructed images using the scale coefficients only (referred to as coarse approximation images).

We emphasize that the goal here is not texture synthesis but rather texture discrimination. In this regard, it can be seen from Figure 6.6 that the extracted 18 GGD model parameters in fact capture some of the main texture-specific information of the indexed images. The synthetic images from two different images using additive GGD models are clearly more distinguishable than the ones using scaling coefficients only.

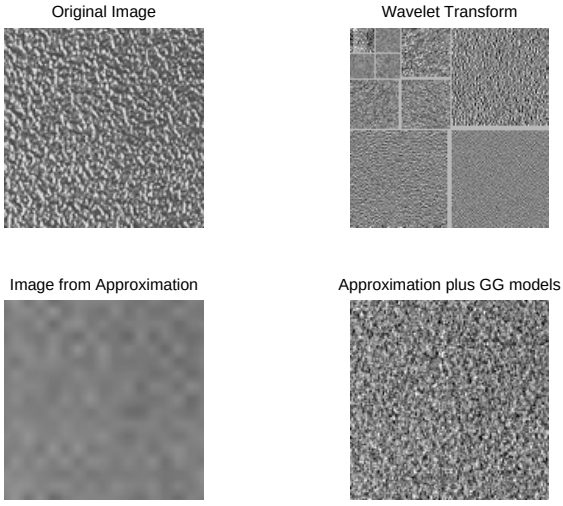
6.4.2 Computational Complexity

The proposed texture retrieval system has been implemented in a Matlab environment. The Feature Extraction (FE) step involves talking a wavelet transform of the input image and estimating the GGD model parameters at each subband using ML estimator. It was found that roughly the same amount of time is spent on wavelet transformation and parameter estimation, giving a total of less than 1 second of CPU time on a Sun's Ultra 5 workstation for extracting features from one image.

We applied three levels of wavelet decomposition (which generates 9 wavelet subbands). Therefore, to represent each texture image using the GGD model we need only 18 numbers as an extracted feature set. Thanks to the closed form of distance in (6.17), the Similarity Measurement (SM) between two im-



(a) Fabric15 subimage.



(b) Metal0 subimage.

Figure 6.6: Texture synthesis examples using generalized Gaussian density for wavelet coefficients on images of size 128×128.

ages involves simple computation using a small number of model parameters. Optimized implementation using lookup tables yield comparable computation time as normalized Euclidean distance.

6.4.3 Range of β in GGD Models

It is of interest to know the common range for the values of β in GGD's for texture images. For typical natural images which are dominated by smooth regions, the values for β are found to be between 0.5 and 1 [102]. Figure 6.7 shows the histogram of the estimated values of β from our database of 640 texture images using the method described in Section 6.3.2. The discrete wavelet transform of three levels using D_4 filters was used. As can be seen from the figure, the 5-th and 95-th percentile values of the estimated β values for our texture database are around 0.7 and 2.0, respectively.

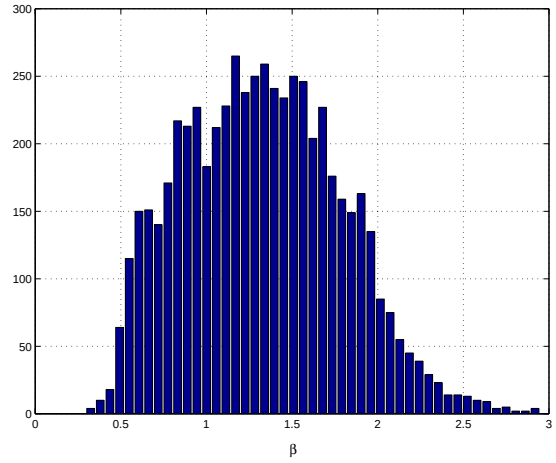


Figure 6.7: Histogram of estimated values for β from 640 texture images of size 128×128.

6.4.4 Retrieval Effectiveness

In retrieval experiments, a simulated query image is any one of 640 images in our database. The relevant images for each query are defined as the other 15 subimages from the same original VisTex image. Following [107] we evaluated the performance in terms of the average rate of retrieving relevant images as a function of the number of top retrieved images. The new approach is compared with the traditional methods using the energy-based features in the wavelet domain given in (6.10-6.11) together with normalized Euclidean distance as the similarity measurement.

Evaluation of all possible wavelet transform schemes is beyond the scope of the experiments. Thus we restricted our attention to the Daubechies family of orthogonal wavelets [42]. This family is indexed by the number of zeros at $\omega = \pi$ of the lowpass filter. Wavelet D_n uses an orthogonal filter bank with length $2n$ lowpass and highpass filters.

For the number of decomposition levels, our experiments agree with [24] that the size of the smallest subimages should not be less than 16×16 so the

estimated energy values or model parameters would be robust. Hence for the input image size 128×128, a maximum of 3 levels of decomposition is chosen.

Filters	Methods				
	$\mathbf{L}^1$	$\mathbf{L}^2$	$\mathbf{L}^1 + \mathbf{L}^2$	GGD & KLD	GGD & ED
D_2	55.50	53.37	56.89	**68.36**	58.73
D_4	55.02	51.37	55.47	**66.36**	57.47
D_6	55.39	53.49	56.15	**65.60**	56.41
D_8	55.80	54.37	56.54	**65.37**	56.45
D_{10}	55.04	52.71	54.97	**64.22**	56.10

(a) One-level decomposition.

Filters	Methods				
	$\mathbf{L}^1$	$\mathbf{L}^2$	$\mathbf{L}^1 + \mathbf{L}^2$	GGD & KLD	GGD & ED
D_2	61.29	57.54	61.43	**74.10**	64.63
D_4	62.97	59.68	62.82	**73.35**	65.87
D_6	61.88	59.57	61.65	**72.72**	64.89
D_8	62.83	61.25	62.99	**72.65**	65.15
D_{10}	62.21	60.62	62.11	**71.61**	63.45

(b) Two-level decomposition.

Filters	Methods				
	$\mathbf{L}^1$	$\mathbf{L}^2$	$\mathbf{L}^1 + \mathbf{L}^2$	GGD & KLD	GGD & ED
D_2	62.72	61.76	64.48	**76.93**	62.22
D_4	63.89	62.54	64.83	**76.57**	61.18
D_6	65.07	63.03	65.11	**75.51**	60.35
D_8	65.60	63.64	65.48	**75.63**	61.73
D_{10}	65.90	64.25	65.81	**74.66**	59.73

(c) Three-level decomposition.

Table 6.1: Average retrieval rate (%) in the top 15 matches using pyramid wavelet transform with different filters and decomposition levels.

In a first series of experiments, the wavelet pyramid transform (DWT) with different filters and decomposition levels were employed. Table 6.1 shows the comparison in performance in average percentages of retrieving relevant images in the top 15 matches. Here $\mathbf{L}^1$ and $\mathbf{L}^2$ denote the methods which use texture features computed as in (6.10) and (6.11), respectively; while $\mathbf{L}^1 + \mathbf{L}^2$ uses both set of features. Note that the $\mathbf{L}^1 + \mathbf{L}^2$ feature set is essentially equivalent to the one composed of means and standard deviations of the magnitude of the wavelet coefficients that was used in [107]. It also yields the same number of features per

images as the statistical method using GGD's (2 features per wavelet subband). The proposed method use the combination of GGD & KLD. We also report the results when the normalized Euclidean distance (ED) was used on GGD's model parameters, which is similar to the method used in [179]. Therefore, except GGD & KLD, all other listed methods use normalized Euclidean distance as the similarity measurement. Following are the main points that we observed:

1. First, the statistical approach (GGD & KLD) always outperforms the traditional methods. This is consistent with our expectation since the GGD parameters are more expressive in characterizing textures than the energy-based ones. Furthermore, the inferior results of the GGD & ED method (where the same features with the statistical method were used but with the normalized Euclidean distance) shows that good performance in retrieval comes not just from a good set of extracted features but also together with a suitable similarity measurement. Hence this supports our approach of considering the two problems FE and SM jointly.

2. Secondly, the length of the filter has little effect in performance in all methods.

3. Finally, in our database, most of the texture discrimination information live in the first two scales of wavelet decomposition since there is little improvement in retrieval rates when we increased from 2 to three levels of decomposition.

Figure 6.8 details the comparison between the $\mathbf{L}^1 + \mathbf{L}^2$ and GGD & KLD methods on each texture class using 3 levels of wavelet decomposition with the D_4 filters. Again we can see that the new method consistently gives superior performance for almost all texture classes, especially for the ones that have structural patterns. Note that those two methods extract the same number of features, 18 ($= 2 \times 9$), from each indexed image.

Figure 6.9 shows a graph illustrating this comparison in retrieval performances as functions of number of top matches considered. As can be seen, almost the double number of retrieved images is required in the traditional method compared to the new method in order to retrieve the same number of relevant images.

A second series of experiments was conducted for the non-subsampled discrete wavelet frames (DWF). The results of retrieval rates are summarized in Table 6.2. We also listed the results for the wavelet pyramids (DWT) for comparison. As expected, the full rate filterbanks improve the performance over the critical-sampled filterbanks on both the old and new methods. However this improvement is marginal compared to the one achieved when replacing the traditional approach by our proposed method.

6.4.5 Image Retrieval Examples

Qualitative evaluation of our method was carried out by visually examining the images of retrieval results. However, this can only be based on a subjective

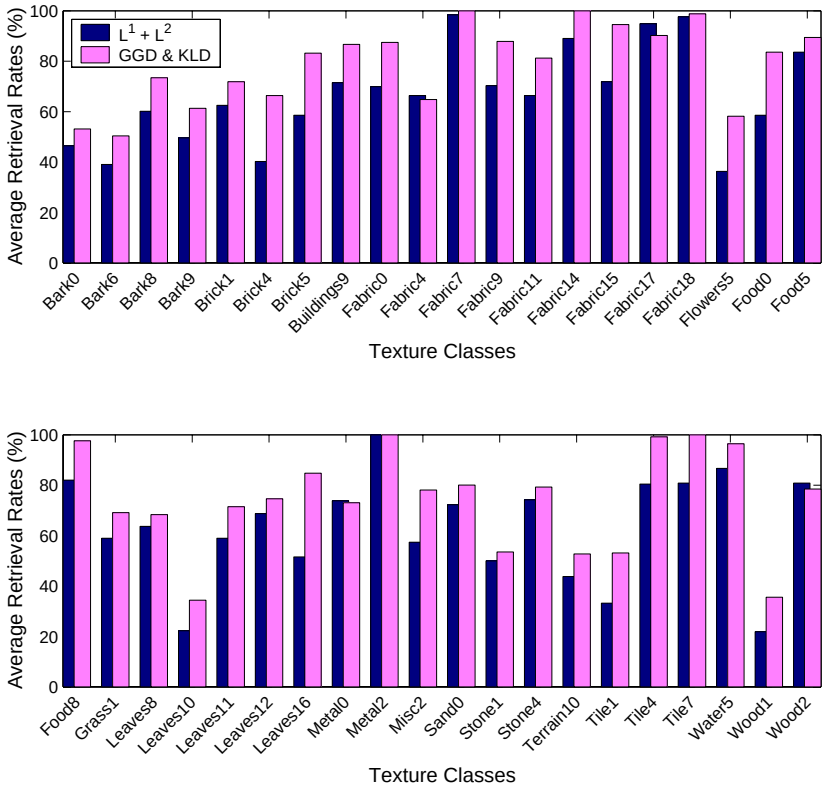


Figure 6.8: Average retrieval rates for individual texture class using wavelet pyramid transform with Daubechies' D_4 filters and three decomposition levels.

Type of decomposition	Methods		
	$\mathbf{L^1 + L^2}$	GGD & KLD	GGD & ED
1 scale (6 features)			
DWT	55.47	**66.36**	57.47
DWF	56.92	**67.09**	61.34
2 scales (12 features)			
DWT	62.82	**73.35**	65.87
DWF	63.32	**74.01**	69.78
3 scales (18 features)			
DWT	64.83	**76.57**	61.18
DWF	68.48	**78.12**	71.13

Table 6.2: Average retrieval rate (%) in the top 15 matches using pyramid wavelet transform (DWT) and wavelet frames (DWF) with D_4 filters.

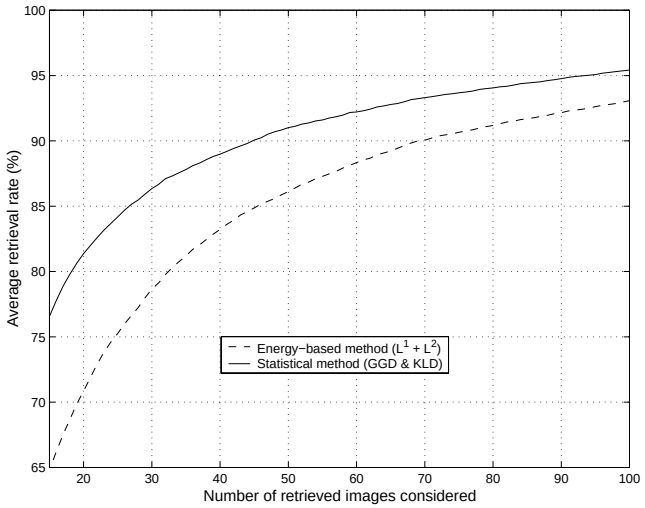
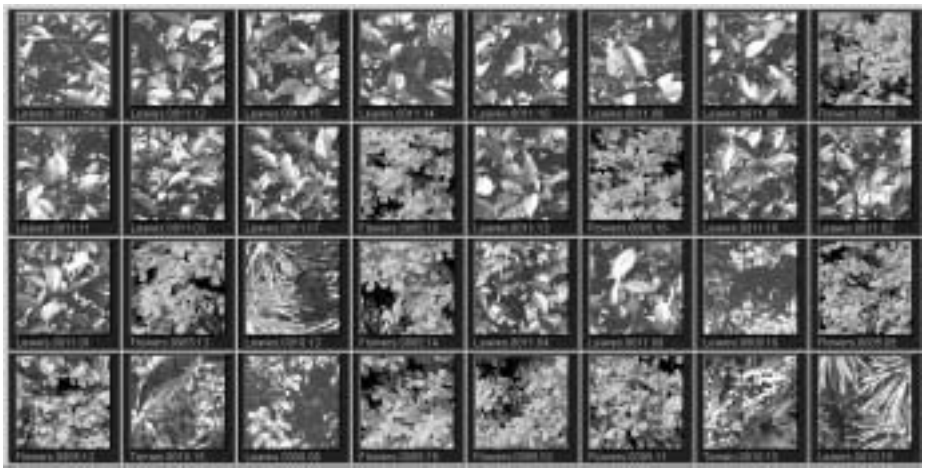


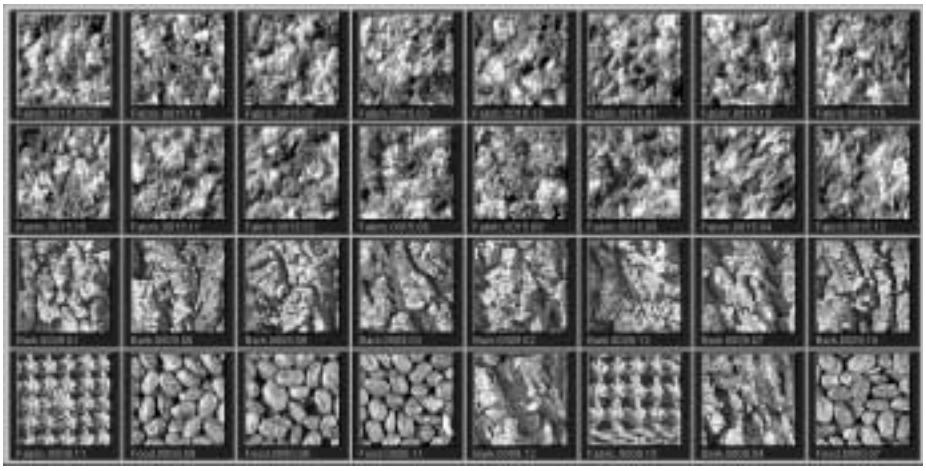
Figure 6.9: Retrieval performance according to the number of top matches considered.

perceptual similarity since there exists no "correct" ordering that is agreed upon by all people [100].

Figure 6.10 shows some examples of retrieval results to demonstrate the capability of our method. In Figure 6.10(a), the query image is "leaves". The system almost perfectly retrieves all images of the same leaves and also images of other types of leaves. In Figure 6.10(b), the query is a fabric patch. In this case, all relevant images are correctly ranked as the top matches following by images of similar textures.



(a)



(b)

Figure 6.10: Examples of retrieval results from 640 texture images based on the VisTex collection. In each case, the query image is on the top left corner; all other images are ranked in the order of similarity with the query image from left to right, top to bottom.

6.5 Wavelet-domain Hidden Markov Models

6.5.1 Overview

While having low complexity, the marginal distribution model ignores some important texture-specific information, notably the dependency of wavelet descriptors across scales and orientations. Furthermore like most other wavelet-based texture analysis methods, the extracted features are sensitive to the orientation of the analyzed image. This is a drawback in the retrieval applications since a same texture can appear at different orientations in the image database.

In this section, we address these problems by using a coherent statistical model that captures both wavelet subband marginal distributions and inter-band dependencies while being rotation invariant. The proposed model uses a wavelet domain hidden Markov tree [38] and steerable pyramids [152]. Rotation invariance is achieved via a diagonalization of the covariance matrices in the model.

First, this work enhances the recent technique on wavelet-domain hidden Markov models (WD-HMM) [38] for better dealing with images by incorporating the dependency of wavelet coefficients across orientations. Second, by replacing the standard wavelet transform with an overcomplete representation via steerable pyramids [151], we obtain a steerable statistical model that can facilitate rotation invariant applications. Finally, although WD-HMM's have been applied in the texture segmentation problem [28], only a small number of texture classes (typically a few regions) were classified. Our experiments with WD-HMM's in the image retrieval application provide a large scale evaluation of their capacity in discriminating among many different texture classes.

Several authors have developed rotation invariant texture features, mostly for the classification problem – where the classes are defined *a priori*. Chen and Kundu [25] modeled the features of wavelet subbands as a hidden Markov model (HMM). These models are trained using texture samples with different orientations that are treated as being in the same class. Greenspan et. al. [74] and Haley and Manjunath [76] used the magnitude of a discrete Fourier transform in the rotation dimension of features obtained from a multiresolution filtering. Wu and Wei [180] achieved rotation invariance by first converting 2-D texture images to 1-D signal via spiral sampling, and then a subband decomposition is applied to the 1-D signal and followed by an HMM on the subband features.

6.5.2 Scalar Model

Recently, Crouse et. al. [38] proposed a new framework for statistical signal processing based on wavelet-domain hidden Markov models (WD-HMM's). It provides an attractive approach to model both the non-Gaussian statistics and the persistence across scale property of wavelet coefficients that are often found in real-world signals. The concept of WD-HMM is briefly reviewed in this section together with the introduction of the notations.

In a WD-HMM, to each wavelet coefficient W_i there is an associated discrete hidden state S_i with the probability mass function $P(S_i = m) = p_i^m$; $m = 1, \ldots, M$. Conditioning on $S_i = m$, W_i follows a Gaussian density with the mean $\mu_{i,m}$ and the variance $\sigma_{i,m}^2$. Since the wavelet coefficients are resulted from convolutions with filters that have zero sum (the wavelet high-pass filters), they can be considered to have zero mean. Furthermore, to reduce the number of parameters in the models, wavelet coefficients at the same subband are *tied* together to share the same statistics. If we take $M = 2$, the marginal distribution wavelets coefficients at the j-th subband can be written as a mixture of two Gaussian densities

$$f_j(w) = p_j^1 g(w; \sigma_{j,1}) + p_j^2 g(w; \sigma_{j,2}), \tag{6.24}$$

where $p_j^1 + p_j^2 = 1$ and $g(w; \sigma)$ denotes the zero-mean Gaussian density with variance σ^2, i.e. $g(w; \sigma) = (2\pi\sigma^2)^{-1/2} \exp(-w^2/2\sigma^2)$.

In this model, p_j^1 and p_j^2 have physical interpretation as the probability that a wavelet coefficient W at the level j is in the state of being small and large, respectively. Small coefficients can be considered as outcomes of a small variance probability density function, whereas large ones can be considered as outcomes of a large variance density.

There is an inter-scale dependency, most notably between a wavelet coefficient at a coarse level (parent) to the four coefficients at the next intermediate level that correspond to the same location (children) in the image (see Figure 6.12(a)). In order to capture this persistence across scales, there are state transition probability matrices for the parent $\rightarrow$ child link between the hidden states

$$\mathsf{A}_j = \begin{bmatrix} p_j^{1\to 1} & p_j^{1\to 2} \\ p_j^{2\to 1} & p_j^{2\to 2} \end{bmatrix}, \qquad j = 2, 3, \ldots, J. \tag{6.25}$$

Here $p_j^{m\to m'}$ is the probability that a child coefficient at the level j will be in state m' given its parent coefficient is in state m. It is clear that A_j has row sums equal to 1.

By denoting $\rho(i)$ the parent of the node i in the wavelet coefficient tree, we have

$$P(S_i = m) = \sum_{m'} P(S_{\rho(i)} = m')\, P(S_i = m | S_{\rho(i)} = m').$$

For a tied WD-HMM (where all the nodes at the same level j share the same statistics) we have

$$p_j^m = \sum_{m'} p_{j-1}^{m'} p_j^{m'\to m}, \qquad j = 2, 3, \ldots, J. \tag{6.26}$$

If we denote $\boldsymbol{p}_j = [p_j^1, p_j^2]$, then (6.26) is equivalent to $\boldsymbol{p}_j = \boldsymbol{p}_{j-1}\mathsf{A}_j$. Hence, we have

$$\boldsymbol{p}_j = \boldsymbol{p}_1 \mathsf{A}_2 \mathsf{A}_3 \ldots \mathsf{A}_j, \qquad \forall j = 2, \ldots, J. \tag{6.27}$$

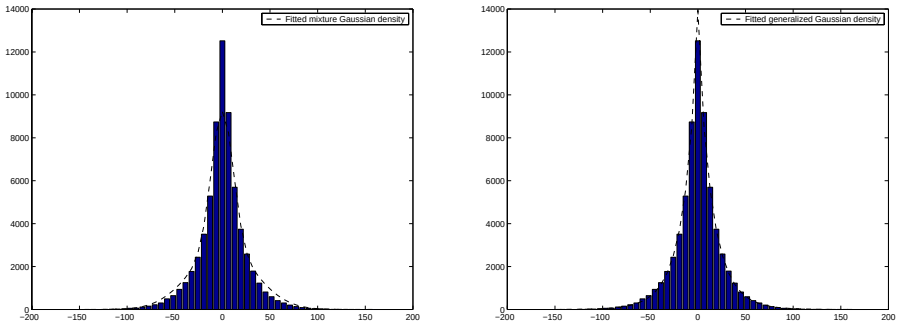
Therefore, the WD-HMM for a tree of wavelet coefficients (also called hidden Markov tree model) is completely defined by a set of model parameters:

$$\Theta = \{\boldsymbol{p}_1, \mathsf{A}_2, \ldots, \mathsf{A}_J;\ \sigma_{j,1}, \sigma_{j,2}\ (j = 1, \ldots, J)\} \tag{6.28}$$

Here J is the number of wavelet tree levels. The result is a statistical model that effectively captures both the marginal and the joint parent-child distributions of wavelet coefficients. Moreover there exists an efficient Expectation Maximization (EM) algorithm for fitting a WD-HMM to observed signal data using the ML criterion [38].

Originally developed for 1-D signals, the WD-HMM has been generalized for images in segmentation [28] and denoising [138] applications. For images, the wavelet transform leads to a decomposition with three orientations, often called horizontal (H), vertical (V) and diagonal (D). The authors in [28, 138] took a simple approach by considering these three orientations separately, thus requiring training three WD-HMM's to characterize an image, one for each orientation. We refer to those models as *scalar* WD-HMM's.

Figure 6.11 shows a typical example of the histogram of the wavelet coefficients from an image subband, together with the plot of the subband marginal density function obtained from the trained WD-HMM. By construction, the estimated marginal density is a mixture of two Gaussian densities as given in (6.24). For comparison we also show the fitted GGD using the ML estimator [60]. As can be seen from the figure, the WD-HMM provides a close match to the GGD in terms of modeling the marginal distribution from a wavelet subband. However, the WD-HMM is more expressive than the GGD model by adding the dependencies between parent-child coefficients across scales.



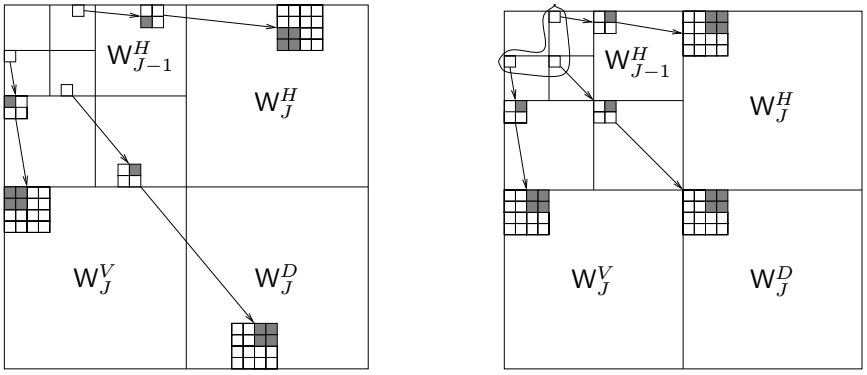
(a) WD-HMM. (b) GGD.

Figure 6.11: Example of wavelet subband coefficient histogram fitted with the marginal distribution curves by WD-HMM and GGD model.

6.5.3 Vector Model

The underlying assumption for the scalar WD-HMM approach is that wavelet coefficients at different orientations are independent. However, experiments in [153] show the importance of the cross-correlation of each subband with other orientations at the same scale in characterizing texture images.

To enhance the capacity of WD-HMM in capturing the cross-orientation dependency of wavelet coefficient, we propose to group coefficients at the same location and scale into vectors and then model them by a single multidimensional WD-HMM (see Figure 6.12(b)). The result is one *vector* WD-HMM for the whole input image.



(a) Scalar WD-HMM. (b) Vector WD-HMM.

Figure 6.12: Tree structures on the WD-HMM's. In the scalar WD-HMM there are three scalar models whereas in the vector WD-HMM there is one vector model. (a) Scalar model. (b) Vector model.

More specifically, denote the wavelet coefficients at the orientation d ($d = 1, 2, 3$ for H, V, D respectively), scale j and location $\boldsymbol{k}$ as $w^d_{j,\boldsymbol{k}}$. The grouping operation will produce vectors of coefficients $\boldsymbol{w}_{j,\boldsymbol{k}} = [w^1_{j,\boldsymbol{k}}, w^2_{j,\boldsymbol{k}}, w^3_{j,\boldsymbol{k}}]^T$. Note that $\boldsymbol{w}_{j,\boldsymbol{k}}$ can be seen as the result of the inner products between the input image $\boldsymbol{x}$ with the three (local) directional wavelet basis functions $\psi^d_{j,\boldsymbol{k}}$ at scale j and location $\boldsymbol{k}$ [103],

$$w^d_{j,\boldsymbol{k}} = \langle \boldsymbol{x}, \psi^d_{j,\boldsymbol{k}} \rangle, \qquad d = 1, 2, 3.$$

The marginal distribution function of the wavelet coefficient vectors at the level j in the vector WD-HMM (with tying) is expressed as

$$f_j(\boldsymbol{w}) = p^1_j g(\boldsymbol{w}; \mathsf{C}_{j,1}) + p^2_j g(\boldsymbol{w}; \mathsf{C}_{j,2}). \tag{6.29}$$

Here $g(\boldsymbol{w}; \mathsf{C})$ denotes the zero-mean multivariate Gaussian density with co-

variance matrix C, i.e.

$$g(\boldsymbol{w};\mathsf{C}) = \frac{1}{\sqrt{(2\pi)^n |det(\mathsf{C})|}} \exp(-\boldsymbol{w}^T \mathsf{C}^{-1} \boldsymbol{w}), \tag{6.30}$$

where $n = 3$ is the number of orientations.

The wavelet coefficient vectors are then organized into the same quad-tree structure that connects each vector to its four children at the next intermediate level of the same location (see Figure 6.12(b)). The parent $\rightarrow$ child link relationships of these vectors are captured in the same way as in (6.25) for the scalar WD-HMM. Thus, an image is modeled by one vector WD-HMM with a set of parameters:

$$\Theta = \{\boldsymbol{p}_1, \mathsf{A}_2, \ldots, \mathsf{A}_J;\ \mathsf{C}_{j,1}, \mathsf{C}_{j,2}\ (j = 1, \ldots, J)\}. \tag{6.31}$$

Thus in a vector WD-HMM, wavelet coefficients at the same scale and location but different orientations are tied up to have same hidden state. The justification for this is that around edges in an image, wavelet coefficients at all orientations have high probability of being significant; whereas in smooth regions, all wavelet coefficients are small. Additionally, in the vector WD-HMM, the across orientation dependencies are captured via the non-diagonal entries in the covariance matrices of the multivariate Gaussian densities (6.30).

Since any marginal density from a multivariate Gaussian density is also a Gaussian density, from (6.29), the marginal density for each wavelet subband in a vector WD-HMM is also a mixture of two zero-mean Gaussian densities. Thus, one can expect that the vector WD-HMM also captures well the subband dependent marginal probability distributions of wavelet coefficients as the scalar WD-HMM.

6.5.4 Relations Between Models

In this section we draw the connections between the generalized Gaussian density (GGD) model and the scalar and vector WD-HMM's. As already discussed, all of these models capture very well the subband-dependent marginal probability density function. This is a crucial point since psychological research on human texture perception suggests that two homogeneous textures are often difficult to discriminate if they produce similar marginal distributions of responses from a bank of filters [11].

In [53] by simply modeling those PDF's by GGD's, we have already obtained very good retrieval results, compared to the traditional subband energy approach. The scalar WD-HMM adds on an extra texture-specific information by capturing the inter-scale dependencies (via the A_j matrices). The vector WD-HMM furthermore adds on the inter-orientation dependencies information (via the non-diagonal entries in the covariance matrices) in characterizing textures.

Table 6.3 shows the number of free parameters needed to describe each image using different models, assuming the wavelet transform is decomposed

with J levels. Note that due to the row sums property, each A_j has only 2 free parameters. The covariance matrices C are symmetric, thus they contain 6 free parameters each.

	GGD	scalar WD-HMM	vector WD-HMM
Hidden states	–	$3 \times (2J-1)$	$2J-1$
Densities	$3 \times J \times 2$	$3 \times J \times 2$	$J \times 2 \times 6$
Total	$6J$	$12J-3$	$14J-1$

Table 6.3: Number of free parameters needed to specify different models for an image when the wavelet transform is taken to J levels.

6.5.5 Steerable WD-HMM

Both the scalar and the vector WD-HMM's described above have drawbacks in that they are sensitive to the orientation of the input image. This problem has roots in the standard wavelet transform. If the image is rotated then in the wavelet domain the wavelet coefficients change completely (except for the rotations of $k\pi/2,\ k \in \mathbb{Z}$). In fact, the wavelet coefficients of the rotated image will *not* just be simply rotated, but also will be modified.

One way to remedy this situation is to replace the standard wavelet decomposition with a steerable pyramid [152, 151]. The steerable pyramid is a linear multi-scale, multi-orientation image decomposition where the basis functions are directional derivative operators. This transform satisfies the *shiftability* in orientation condition which states that at a fixed location and scale, one can compute responses at any orientation via a linear combination of coefficients corresponding to the oriented basis functions at that location and scale. As a bonus, the steerable pyramid representation is also translation-invariant.

Denote $\boldsymbol{w}$ and $\boldsymbol{w}^\phi$ as the vectors of steerable pyramid coefficients at fixed scale j and location $\boldsymbol{k}$ for an input image and its rotated copy by ϕ respectively, then we have

$$\boldsymbol{w}^\phi = \mathsf{R}_\phi \boldsymbol{w}. \tag{6.32}$$

The columns of R_ϕ are a set of interpolation functions that depend only on the rotation angle ϕ and the steerable basis functions. Furthermore, orientation shiftability ensures the orientation invariance of response power [152], i.e. $||\boldsymbol{w}^\phi||^2 = ||\boldsymbol{w}||^2$ for any $\boldsymbol{w}$. This is equivalent [80] to saying that R_ϕ is an orthogonal matrix, i.e. $\mathsf{R}_\phi^{-1} = \mathsf{R}_\phi^T$.

The following proposition shows that for the vector WD-HMM on a steerable pyramid, if the input image is rotated then the new model can be obtained by a simple transformation of the original WD-HMM.

Proposition 6.1 *Suppose that* $\Theta = \{\boldsymbol{p}_1, \mathsf{A}_2, \ldots, \mathsf{A}_J;\ \mathsf{C}_{j,i}\ (j = 1, \ldots, J;\ i = 1, 2)\}$ *is the vector WD-HMM on a steerable pyramid for a given homogeneous*

texture class. Then the corresponding model for the rotated version of that texture by ϕ is $\Theta^\phi = \{\boldsymbol{p}_1, \mathsf{A}_2, \ldots, \mathsf{A}_J;\ \mathsf{C}_{j,i}^\phi\ (j = 1, \ldots, J;\ i = 1, 2)\}$. The only change are the covariance matrices

$$\mathsf{C}_{j,i}^\phi = \mathsf{R}_\phi \mathsf{C}_{j,i} \mathsf{R}_\phi^T, \qquad j = 1, \ldots, J;\ i = 1, 2. \tag{6.33}$$

Proof: The marginal distribution function of the coefficient vectors $\boldsymbol{w}^\phi$ at the level j of the rotated texture can be written as

$$\begin{aligned} f_j^\phi(\boldsymbol{w}^\phi) &= \frac{1}{J(\boldsymbol{w}^\phi, \boldsymbol{w})} f(\mathsf{R}_\phi^{-1}\boldsymbol{w}^\phi) \\ &= p_j^1 g(\mathsf{R}_\phi^{-1}\boldsymbol{w}^\phi; \mathsf{C}_{j,1}) + p_j^2 g(\mathsf{R}_\phi^{-1}\boldsymbol{w}^\phi; \mathsf{C}_{j,2}), \end{aligned}$$

since the Jacobian $J(\boldsymbol{w}^\phi, \boldsymbol{w}) = |det(\mathsf{R}_\phi)| = 1$. Using the fact that R_ϕ is an orthogonal matrix again, by manipulating (6.30) we have

$$g(\mathsf{R}_\phi^{-1}\boldsymbol{w}^\phi; \mathsf{C}) = g(\boldsymbol{w}^\phi; \mathsf{R}_\phi \mathsf{C} \mathsf{R}_\phi^T).$$

Thus $f_j^\phi(\boldsymbol{w}^\phi)$ is also a mixture of two zero-mean multivariate Gaussian densities which has the same probability mass function p_j^m for the hidden state as in $f_j(\boldsymbol{w})$ whereas the covariance matrices are related by Equation (6.33). Combining this across scales we obtain the desired result. □

As a result, the vector WD-HMM on a steerable pyramid is a *steerable* model. In other words, one can train a WD-HMM for a single orientation of a texture and then steer this model to describe that texture at any other orientation.

6.5.6 Rotation Invariance using Steerable WD-HMM

Using the steerable WD-HMM above, we now develop a rotation-invariant model for texture retrieval. Recall that the only difference between the steerable WD-HMM's Θ and Θ^ϕ of a given texture and its rotated version are among the covariance matrices. These covariance matrices are related by (6.33), or $\mathsf{C}_{j,i}$ and $\mathsf{C}_{j,i}^\phi$ are said to be orthogonally equivalent [80].

Using the Takagi's factorization [80], we factor each covariance matrix in the steerable WD-HMM into a product

$$\mathsf{C}_{j,i} = \mathsf{U}_{j,i} \Lambda_{j,i} \mathsf{U}_{j,i}^T, \qquad j = 1, \ldots, J;\ i = 1, 2, \tag{6.34}$$

where $\mathsf{U}_{j,i}$ is the orthogonal matrix whose columns are the normalized eigenvectors of $\mathsf{C}_{j,i}$ and $\Lambda_{j,i}$ is the diagonal matrix containing the real, nonnegative eigenvalues of $\mathsf{C}_{j,i}$ in descending order. This factorization is always possible since all covariance matrices are symmetric and positive semidefinite.

Let λ be an eigenvalue of $\mathsf{C}_{j,i}^\phi$. That means there exists a vector $\boldsymbol{v}$ satisfies $\mathsf{C}_{j,i}^\phi \boldsymbol{v} = \lambda \boldsymbol{v}$. Using (6.33) we have

$$\mathsf{R}_\phi \mathsf{C}_{j,i} \mathsf{R}_\phi^T \boldsymbol{v} = \lambda \boldsymbol{v} \qquad \text{or} \qquad \mathsf{C}_{j,i} \mathsf{R}_\phi^T \boldsymbol{v} = \lambda \mathsf{R}_\phi^T \boldsymbol{v}.$$

So if we let $\boldsymbol{v}' = \mathsf{R}_\phi^T \boldsymbol{v}$ then $\mathsf{C}_{j,i}\boldsymbol{v}' = \lambda \boldsymbol{v}'$ hence λ is also an eigenvalue of $\mathsf{C}_{j,i}$. Thus the diagonalization operation on the rotated model Θ^ϕ leads to

$$\Lambda_{j,i}^\phi = \Lambda_{j,i}, \qquad \forall j = 1, \ldots, J;\ i = 1, 2.$$

In summary, given a steerable WD-HMM, if we factorize the covariance matrices into the form (6.34) then the $\mathsf{U}_{j,i}$ matrices are responsible for the orientation of the input image while the $\Lambda_{j,i}$ matrices contain rotation-invariant texture information. Thus we have the following result.

Proposition 6.2 *The* diagonalized *steerable WD-HMM*

$$\bar{\Theta} = \{\boldsymbol{p}_1, \mathsf{A}_2, \ldots, \mathsf{A}_J;\ \Lambda_{j,i}\ (j = 1, \ldots, J;\ i = 1, 2)\} \tag{6.35}$$

is a rotation-invariant model.

Remark 6.1 *In practice one estimates a WD-HMM for an input image via the EM algorithm using the ML criterion. So the rotation invariant property of the estimated model relies on the assumption that the ML solution of the WD-HMM is unique and the EM training algorithm is able to find it.*

6.6 Experiments with Wavelet-domain HMM's

6.6.1 Databases

We use two texture databases in our experiments. In a first series of experiments, we evaluate the retrieval effectiveness of both scalar and vector WD-HMM against the GGD method in a large database. For this, we employed the same test database of 640 texture images obtained from the VisTex collection that has been used in Section 6.4.

The second image collection is used to test the rotation-invariant property of WD-HMM's. It consists of thirteen 512×512 Brodatz texture images that were rotated to various degrees *before* being digitized [164]. Figure 6.13 displays the original textures at the 0 degree or non-rotated position. From these images, we first construct the *non-rotated* image set by dividing each of the original 0 degree image into sixteen 128×128 non-overlapping subimages. Next, we construct the *rotated* image set by taking four non-overlapping 128×128 subimages each from the original images at 0, 30, 60, and 120 degrees. Both databases contain 208 images that come from 13 texture classes. The *non-rotated* set serves as the ideal case, where all images in a same class have same orientation, for the *rotated* set.

As before, in our retrieval experiments, each image in the database is simulated as a query. The relevant images for each query are defined as the other 15 subimages from the same original 512×512 image. The retrieval performance is evaluated in term of the percentage of relevant images among the top 15 retrieved images.

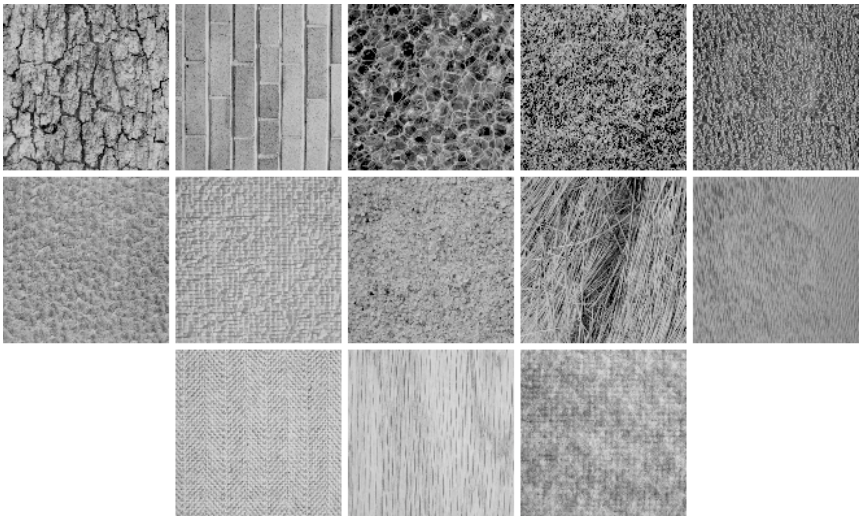


Figure 6.13: Texture images from the rotate collection that are used in the experiments. The images are at 0° degree. From left to right and top to bottom are: Bark, Brick, Bubbles, Grass, Leather, Pigskin, Raffia, Sand, Straw, Water, Weave, Wood, and Wool.

6.6.2 Kullback-Leibler Distance between WD-HMM's

The statistical framework in Section 6.2 suggests that the Kullback-Leibler distance (KLD) should be used to compute the dissimilarity between WD-HMM's. An additional advantage of using the KLD is that since it is defined directly on the extracted model's parameters, then with rotation-invariant models it leads to a rotation-invariant image retrieval system. However, there is no closed form expression the KLD between hidden Markov models.

A simple solution is to resort to a Monte-Carlo method for computing the integral in the KLD. More specifically, from the query model we randomly generate data as trees of wavelet coefficients, and then compute its likelihood against each candidate models. Figure 6.14 shows the computed KLD's between two models using Monte-Carlo method according to the number of randomly generated trees used to compute the distance. The KLD converges rapidly and the variance decreases as the number of used random trees increases. In our experiments we noticed that beyond 64 randomly generated trees, the results did not vary much any longer.

In [49], we develop a fast algorithm to approximate of the KLD for dependence tree and hidden Markov models. The computational cost of this algorithm is equal to the Monte-Carlo method that uses only one randomly generated tree. The detail of the algorithm is described in Appendix 6.B

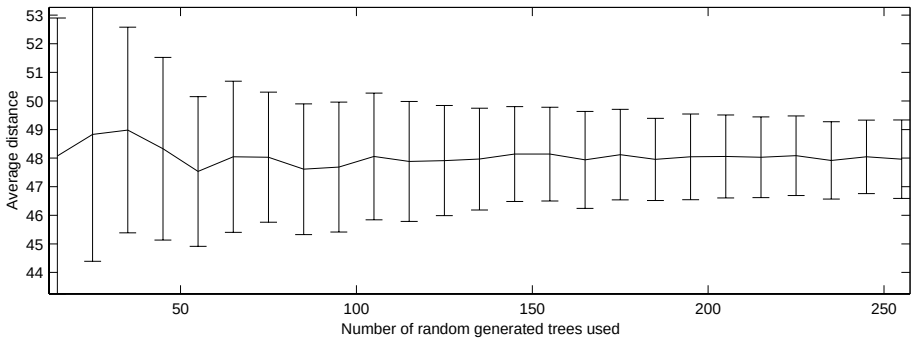


Figure 6.14: Average estimations of the KLD between "Leather" and "Raffia" using the Monte-Carlo method. The error bars are computed as the standard deviation.

6.6.3 Effectiveness of WD-HMM's

For this series of experiments, we used the standard discrete wavelet transform (DWT) with Haar filters and three decomposition levels. We chose three levels of decomposition for our experiments since most of the texture information of our database is concentrated in those three levels.

Figure 6.15 details the retrieval performance on the database of 640 textures images from the VisTex collection by using GGD, scalar WD-HMM and vector WD-HMM to characterize wavelet coefficients. We notice that the scalar WD-HMM method gives compatible result with the GGD method, whereas the vector WD-HMM method significantly improves the retrieval rates in many texture classes, as well on average (Table 6.4). If we focus on the WD-HMM methods, we can see that the vector model almost always outperforms the scalar one. This is consistent with the argument in Section 6.5.3: the vector parameters are more precise in characterizing textures since they include the inter-orientation dependency information.

GGD	scalar WD-HMM	vector WD-HMM
75.73	76.51	80.05

Table 6.4: Average retrieval rates over the whole database for different methods in Figure 6.15.

The fact that the scalar WD-HMM does not provide significant improvement over the GGD as the vector WD-HMM could be explained as follows. First, for most of our test textures, the inter-orientation dependency is more important that the inter-scale dependency across wavelet subbands. Second, the inter-scale dependency captured by a WD-HMM implicitly via hidden states is not expressive enough.

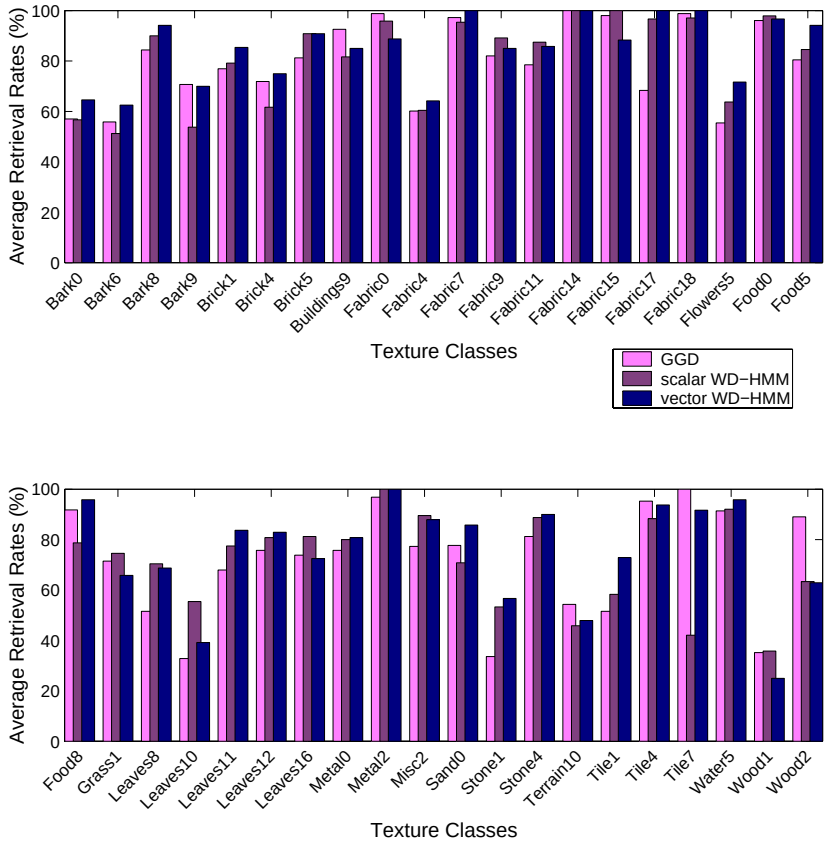


Figure 6.15: Average retrieval rates for individual texture class using standard wavelet transform with Haar filters and 3 decomposition levels.

6.6.4 Effectiveness of Rotation Invariance

In the last series of experiments, we test the rotation invariant property of the steerable WD-HMM as described in Section 6.5.6. We use steerable pyramid having two directions and three levels of decomposition.

Figure 6.16 shows comparison of the performances in average percentages of retrieving relevant images for the *non-rotated* set, the *rotated* set **without** using rotation invariant model, and the *rotated* set **with** rotation invariant model.

First, we compare the retrieval results obtained for the *non-rotated* set against for the *rotated* set **without** using rotation invariance. We see that textures which have similar results for both sets (Bark, Bubbles, Grass, Weave) are the ones that have no strong direction, as those textures are not affected by rotation. Moreover they all have very distinct texture patterns. Textures which are most seriously affected by rotation (Brick, Leather, Pigskin, Raffia, Straw, Water, Wood, Wool) are the ones that are strongly directional.

By contrast, the retrieval results obtained for the *rotated* set **with** rotation invariance are almost the same as those from the *non-rotated* set. Thus, the rotation invariant model is indeed insensitive to the orientation of the analyzed image. The results obtained by exploiting the rotation invariance are very conclusive. The average retrieval rate for the *rotated* set improves by 36.68% when the rotation invariance is effective. The improvement is more striking for the strong directional textures.

However, it is worthwhile to note some details. For the textures Bark and Weave, we notice that the results obtained for the *rotated* image without rotation invariance are better than those of the *non-rotated* set. The reason for this is that the original images for those textures are non-homogeneous. Therefore, when the images are divided into subimages, it can happen that those subimages have different directions. This can be easily observed by looking at Straw texture for instance. In this case, the *rotated* set **with** rotation invariance does not suffer from those differences in direction, therefore the results are better.

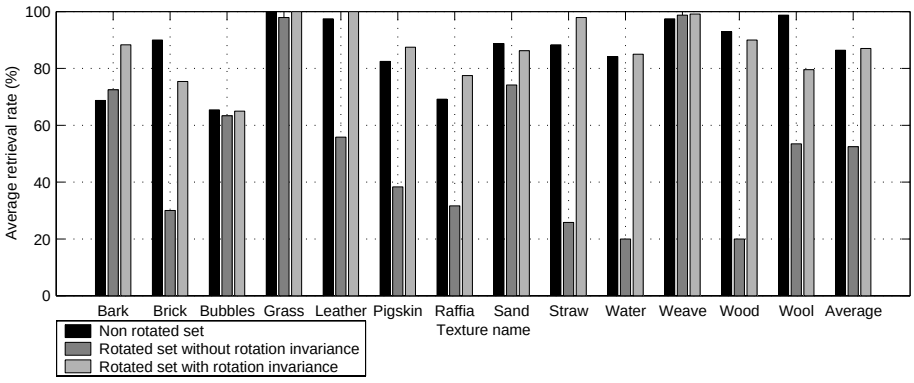


Figure 6.16: Average retrieval rates for individual texture class using order one steerable filters and 3 decomposition levels for *non-rotated* set, *rotated* set **without** rotation invariance and *rotated* set **with** rotation invariance.

6.7 Conclusion and Discussion

We have introduced a statistical framework for texture retrieval in CBIR applications by jointly considering the two problems of FE and SM while keeping in mind the complexity constraint of such applications. In our approach, the FE step becomes an ML estimator for the model parameters of image data and the SM step amounts to computing the Kullback-Leibler distances between model parameters. For large data sets, this achieves the same optimality as the ML selection rule.

The statistical framework has been applied successfully in a wavelet-based texture retrieval application, where wavelet coefficients in each subband are independently modeled by a generalized Gaussian density (GGD). This results

in a new texture similarity measurement in wavelet domain which has a sound theoretical justification with no need for normalization steps. Furthermore, by restricting to simpler models, the new similarity distance becomes closely related to the popular variance-normalized Euclidean distance. Hence, the statistical approach can be used as a common framework for other existing methods. Experimental results on 640 texture images of 40 classes from the VisTex collection indicated that the new method significantly improves retrieval rates, e.g. from 65% to 77%, over the traditional approaches, using both the pyramid wavelet transform and wavelet frames, while requiring comparable computational time.

We then considered more complex texture models that aggregate wavelet descriptors across subbands using hidden Markov models. These models are shown to capture well both the subband marginal distributions and the dependencies of wavelet coefficients across scales and orientations. Furthermore, we propose a new model that can be diagonalized to be rotation invariant. This is a useful property not just for the image retrieval application but also for many other applications. Experimental results indicate that the wavelet-domain hidden Markov models lead to better texture retrieval performance than the independent subband model. On the other hand, it possesses higher computational cost. The rotation invariant property was also tested and results consistent with the theory were obtained.

We want to emphasize that our methods are specially designed for the retrieval problem where the classes are not defined *a priori*. If one considers a pure classification problem, better results might be obtained by taking into account the distribution (e.g. covariances) of the feature vector itself from each predefined class and then employing the optimal Bayesian classifier [165]. Of course, this requires an additional training step which one cannot usually afford in the general retrieval problem.

The proposed statistical framework can be applied to more general retrieval methods. The GGD was used effectively here for modeling the coefficients from the wavelet transforms and wavelet frames and can be applied to other similar filtering schemes such as wavelet packets and Gabor transforms. Furthermore, we can extend the statistical model for texture using the Wold theory which closely matches human texture perception [100]. As shown in Section 6.2.2, the popular histogram method fits into our scheme. Thus besides texture, color and local shape features can also be captured. Finally, assuming that different feature sets (color, texture, shape) to be independent, the chain rule of the KLD suggests that the overall similarity measurement is simply the sum of KLD's from each feature.

Appendix 6.A Maximum Likelihood Estimator for Generalized Gaussian Density

The MLE for GGD amounts to solve the highly nonlinear equation (6.16). We thus have to resort to iterative root finding procedures like the Newton-Raphson

method.

Define the left hand side of (6.16) as a function of $\hat{\beta}$, $g(\hat{\beta})$. The Newton-Raphson iteration finds the new guess for the root of $g(\beta)$, β_{k+1}, based on the previous one, β_k, using

$$\beta_{k+1} = \beta_k - \frac{g(\beta_k)}{g'(\beta_k)} \tag{6.36}$$

We have,

$$\begin{aligned} g'(\beta) &= -\frac{\Psi(1/\beta)}{\beta^2} - \frac{\Psi'(1/\beta)}{\beta^3} - \frac{\sum_{i=1}^{L} |x_i|^\beta (\log|x_i|)^2}{\sum_{i=1}^{L} |x_i|^\beta} + \frac{\left(\sum_{i=1}^{L} |x_i|^\beta \log|x_i|\right)^2}{\left(\sum_{i=1}^{L} |x_i|^\beta\right)^2} \\ &\quad + \frac{1}{\beta^2} + \frac{\sum_{i=1}^{L} |x_i|^\beta \log|x_i|}{\beta \sum_{i=1}^{n} |x_i|^\beta} - \frac{\log\left(\frac{\beta}{L}\sum_{i=1}^{L} |x_i|^\beta\right)}{\beta^2} \end{aligned} \tag{6.37}$$

where $\Psi'(z)$ is known as the first polygamma or trigamma function [1]. Note the fact that $g(\beta)$ and $g'(\beta)$ share many common terms which can be used for saving computation at each iteration step in (6.36).

A good initial guess for the root of $g(\beta)$ can be found based on the matching moments of the data set with those of the assumed distribution [148]. For a GGD, it can be shown that the ratio of mean absolute value to stand deviation is a steadily increasing function of the β:

$$\mathcal{F}_M(\beta) = \frac{\Gamma(2/\beta)}{\sqrt{\Gamma(1/\beta)\Gamma(3/\beta)}} \tag{6.38}$$

Hence if let $m_1 = (1/L)\sum_{i=1}^{L} |x_i|$ and $m_2 = (1/L)\sum_{i=1}^{L} x_i^2$ be the estimate of the mean absolute value and the estimate of the variance of the sample data set, respectively, then β is estimated by solving

$$\bar{\beta} = \mathcal{F}_M^{-1}\left(\frac{m_1}{\sqrt{m_2}}\right) \tag{6.39}$$

In a practical implementation, the solution of (6.39) can be found quickly using interpolation and a look-up table whose entries are the corresponding values of $m_1/\sqrt{m_2}$ and $\bar{\beta}$.

Finally, the initial guess $\beta_0 = \bar{\beta}$ of the ML estimator $\hat{\beta}$ can be "polished up" with a few number steps of Newton-Raphson (experiments showed that only around 3 steps are adequate).

Appendix 6.B Fast Approximation of Kullback-Leibler Distance for Dependence Trees and Hidden Markov Models

Hidden Markov models (HMM) and their generalized versions on dependence tree structure [140], such as hidden Markov trees (HMT), have become powerful

tools in speech recognition [132] and signal processing [38]. Their successes based on their effectiveness in capturing a large classes of natural measurements using a small set of parameters. Furthermore, there are fast algorithms to evaluate and train the models for given data set.

For certain problems, one would like to measure the distances between two statistical models. This measurement is useful in evaluating the training algorithm or classifying the estimated models [85]. The *Kullback-Leibler distance* (KLD) or the *relative entropy* arises in many contexts as an appropriate measurement of the distance between two models. The KLD between the two probability density functions f and $\tilde{f}$ is defined as [37]:

$$D(f\|\tilde{f}) = \int f \log \frac{f}{\tilde{f}}. \tag{6.40}$$

Since for HMM and HMT, the probability function is very complex which can be viewed as a mixture of large number of densities, it can be practically computed via an iterated a recursive procedure (the forward-backward or upward-downward algorithms [132, 140]), and there is no simple closed form expression for the KLD. Typically, the Monte-Carlo method is used to numerically estimate integral in (6.40). This is done by rewriting (6.40) as

$$D(f\|\tilde{f}) = E_f[\log f(X) - \log \tilde{f}(X)].$$

Thus one can randomly generate a sequence of data $\langle x_1, x_2, \ldots, x_T\rangle$ based on the model density $f(X)$ and then approximate the KLD by:

$$D(f\|\tilde{f}) \approx \frac{1}{T}\sum_{n=1}^{T}[\log f(x_n) - \log \tilde{f}(x_n)] \tag{6.41}$$

Obviously, for an accurate approximation of $D(f\|\tilde{f})$, T has to be large, which can be overly expensive in some applications. Furthermore, due to the "randomness" nature of the Monte-Carlo method, the approximations of the distance could vary in different computations. In this section, we present a fast algorithm to approximate the KLD between two dependent tree models or two hidden Markov models. In fact, the algorithm computes an upper bound for the KLD. The complexity of the proposed algorithm is the same as computing *one* log-likelihood in (6.41), thus it is much faster compared with the Monte-Carlo method.

6.B.1 KLD between Dependence Tree Models

Denote $D(\boldsymbol{w}\|\tilde{\boldsymbol{w}})$ as the KLD between two probability mass functions $\boldsymbol{w} = (w_i)_i$ and $\tilde{\boldsymbol{w}} = (\tilde{w}_i)_i$:

$$D(\boldsymbol{w}\|\tilde{\boldsymbol{w}}) = \sum_i w_i \log \frac{w_i}{\tilde{w}_i}. \tag{6.42}$$

Our results based on the following key lemma which was proved for mixture of Gaussians in [154].

Lemma 6.1 *The KLD between two mixture densities* $\sum_i w_i f_i$ *and* $\sum_i \tilde{w}_i \tilde{f}_i$ *is upper bounded by*

$$D\left(\sum_i w_i f_i \| \sum_i \tilde{w}_i \tilde{f}_i\right) \leq D(\boldsymbol{w}\|\tilde{\boldsymbol{w}}) + \sum_i w_i D(f_i\|\tilde{f}_i), \tag{6.43}$$

with equality if and only if $\frac{w_i f_i}{\tilde{w}_i \tilde{f}_i} = const.$

Proof:

$$\begin{aligned} D\left(\sum_i w_i f_i \| \sum_i \tilde{w}_i \tilde{f}_i\right) &= \int \left(\sum_i w_i f_i\right) \log \frac{\sum_i w_i f_i}{\sum_i \tilde{w}_i \tilde{f}_i} \\ &\leq \int \sum_i w_i f_i \log \frac{w_i f_i}{\tilde{w}_i \tilde{f}_i} \\ &= \sum_i w_i \log \frac{w_i}{\tilde{w}_i} + \sum_i w_i \int f_i \log \frac{f_i}{\tilde{f}_i} \\ &= D(\boldsymbol{w}\|\tilde{\boldsymbol{w}}) + \sum_i w_i D(f_i\|\tilde{f}_i), \end{aligned}$$

where the inequality comes from the *log sum inequality* [37]. □

Consider a statistical dependence tree T where at each node n in the tree there is an (hidden) state variable S_n and an observation variable O_n which can be either scalar or vector (Figure 6.17). Denote $\rho(n)$ the parent of the node n and $C(n)$ is the set of children nodes of the node n. Furthermore, denote T_n be the subtree of all nodes with root at n and O_{T_n} is the set of all observation variables attached to these nodes. For convenience, node 1 is assigned to the root of T, thus $T_1 = T$.

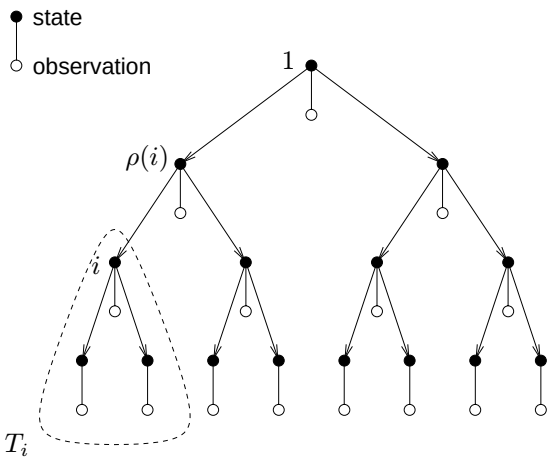


Figure 6.17: A hidden Markov tree model.

The state variables have discrete value in the set $\{1, 2, \ldots, J\}$ and follow a Markov model where the state transition probability is defined on the edges of T as

$$\begin{aligned} P(S_1 = i) &= \pi_i \\ P(S_n = j | S_{\rho(n)} = i) &= a_{ij}^{(n)}. \end{aligned} \tag{6.44}$$

The observation variables have emission probabilities conditioned only on the state of the same node

$$P(O_n = o | S_n = i) = b_i^{(n)}(o), \tag{6.45}$$

where $b_i^{(n)}(.)$ can be either a probability mass function (pmf) for the discrete models or a parameterized probability density function (pdf), typically a mixture of Gaussians, for the continuous models. Therefore the set of parameters $\theta = \left\{\pi_i, a_{ij}^{(n)}, b_i^{(n)}(.)\right\}_{1 \le i,j \le J,\ n \in T}$ completely specifies a hidden Markov tree model.

At each node n, define $\beta_i^{(n)}$ to be the conditional probability density of the subtree observation data which has root at the node n given its state is i, that is

$$\beta_i^{(n)}(o_{T_n}) = P(O_{T_n} = o_{T_n} | S_n = i, \theta), \quad i = 1, \ldots, J. \tag{6.46}$$

For a leaf node n, we have

$$\beta_i^{(n)}(o_n) = b_i^{(n)}(o_n). \tag{6.47}$$

For an intermediate node n, from the definition of HMT model it follows that given the state $S_n = i$, the observation O_n and its subtrees O_{T_m} for each $m \in C(n)$ are independent (refer to Figure 6.17). This leads to the following key induction relation

$$\beta_i^{(n)}(o_{T_n}) = b_i^{(n)}(o_n) \prod_{m \in C(n)} \sum_{j=1}^{J} a_{ij}^{(m)} \beta_j^{(m)}(o_{T_m}) \tag{6.48}$$

This equation is the heart of the "upward algorithm" [140] in which the probabilities β's are computed iteratively up the tree to the root where the probability of the whole observation tree is defined as

$$P(O_T = o_T | \theta) = \sum_{j=1}^{J} \pi_j \beta_j^{(1)}(o_T). \tag{6.49}$$

We next describe an efficient algorithm to approximate the KLD between two HMT models θ and $\tilde{\theta}$.

1. **Initialization:** At each each leaf node n of T, using (6.47) we have

$$D(\beta_i^{(n)} \| \tilde{\beta}_i^{(n)}) = D(b_i^{(n)} \| \tilde{b}_i^{(n)}). \tag{6.50}$$

For discrete models, $D(b_i^{(n)}\|\tilde{b}_i^{(n)})$ can be computed directly as shown in (6.42) for the KLD between two pmf's. For continuous models, where $b_i^{(n)}$ and $\tilde{b}_i^{(n)}$ are mixtures of Gaussians then we can upper bound their KLD using Lemma 6.1 and the following closed form expression for the KLD between two d-dimensional Gaussians [154]:

$$D(\mathcal{N}(.;\boldsymbol{\mu},\mathsf{C})\|\mathcal{N}(.;\tilde{\boldsymbol{\mu}},\tilde{\mathsf{C}})) = \frac{1}{2}[\log\frac{|\tilde{\mathsf{C}}|}{|\mathsf{C}|} - d + \mathrm{trace}(\tilde{\mathsf{C}}^{-1}\mathsf{C}) + (\boldsymbol{\mu}-\tilde{\boldsymbol{\mu}})^T\tilde{\mathsf{C}}^{-1}(\boldsymbol{\mu}-\tilde{\boldsymbol{\mu}})]. \quad (6.51)$$

2. **Induction:** Applying the chain rule of KLD for independent data set $\{O_n, O_{T_m} : m \in C(n)\}$ and Lemma 6.1 to (6.48) we obtain

$$\begin{aligned} D(\beta_i^{(n)}\|\tilde{\beta}_i^{(n)}) &= D(b_i^{(n)}\|\tilde{b}_i^{(n)}) + \sum_{m\in C(n)} D\left(\sum_{j=1}^J a_{ij}^{(m)}\beta_j^{(m)}\|\sum_{j=1}^J \tilde{a}_{ij}^{(m)}\tilde{\beta}_j^{(m)}\right) \\ &\le D(b_i^{(n)}\|\tilde{b}_i^{(n)}) + \\ &\quad \sum_{m\in C(n)}\left(D(\boldsymbol{a}_i^{(m)}\|\tilde{\boldsymbol{a}}_i^{(m)}) + \sum_{j=1}^J a_{ij}^{(m)} D(\beta_j^{(m)}\|\tilde{\beta}_j^{(m)})\right) \quad (6.52) \end{aligned}$$

with equality if and only if $a_{ij}^{(m)}\beta_j^{(m)} = C^{(m)}\tilde{a}_{ij}^{(m)}\tilde{\beta}_j^{(m)}$ for all j, where $C^{(m)}$ is a constant at node m. Here we denote $\boldsymbol{a}_i^{(m)} = (a_{i1}^{(m)},\ldots,a_{iJ}^{(m)})$, which is the pmf for the state of child note S_m given its parent state $S_n = i$.

3. **Termination:** Finally, applying Lemma 6.1 to (6.49)

$$D(\theta\|\tilde{\theta}) \le D(\boldsymbol{\pi}\|\tilde{\boldsymbol{\pi}}) + \sum_{j=1}^J \pi_j D(\beta_j^{(1)}\|\tilde{\beta}_j^{(1)}) \quad (6.53)$$

with equality if and only if $\pi_j\beta_j^{(1)} = C^{(1)}\tilde{\pi}_j\tilde{\beta}_j^{(1)}$ for all j.

In effect, we have a fast algorithm to compute an upper bound for the KLD between two HMT models. This algorithm has the same complexity as the upward algorithm.

6.B.2 KLD between Hidden Markov Models

A special case of HMT models is the hidden Markov model (HMM) which was shown to be useful in many applications, especially speech recognition [132]. In a HMM, the dependence tree T becomes a chain, that is except the last one, each node has exactly one child. Furthermore, all the nodes share the same statistics: $a_{ij}^{(n)} = a_{ij}, b_i^{(n)} = b_i$, for all n. Number the nodes in the dependence chain in the obvious way: start from 1 and end at N for the last node. The inductive relation (6.52) becomes

$$D(\beta_i^{(n)}\|\tilde{\beta}_i^{(n)}) \le D(b_i\|\tilde{b}_i) + D(\boldsymbol{a}_i\|\tilde{\boldsymbol{a}}_i) + \sum_{j=1}^J a_{ij} D(\beta_j^{(n+1)}\|\tilde{\beta}_j^{(n+1)}). \quad (6.54)$$

Denote $d_i = D(\boldsymbol{a}_i \| \tilde{\boldsymbol{a}}_i) + D(b_i \| \tilde{b}_i)$, $D_i^{(n)} = D(\beta_i^{(n)} \| \tilde{\beta}_i^{(n)})$, $\boldsymbol{d} = (d_1, \ldots, d_J)^T$, $\boldsymbol{D}^{(n)} = (D_1^{(n)}, \ldots, D_J^{(n)})^T$, and $\mathsf{A} = \{a_{ij}\}$ be the state-transition probability matrix. Then (6.54) can be written in a more compact form as

$$\boldsymbol{D}^{(n)} \leq \boldsymbol{d} + \mathsf{A}\boldsymbol{D}^{(n+1)}. \tag{6.55}$$

With the obvious same notation, the initialization step (6.50) becomes

$$\boldsymbol{D}^{(N)} = (D(b_1 \| \tilde{b}_1), \ldots, D(b_J \| \tilde{b}_J))^T \stackrel{\text{def}}{=} \boldsymbol{e}.$$

And the termination step (6.53) becomes

$$D(\theta \| \tilde{\theta}) \leq D(\boldsymbol{\pi} \| \tilde{\boldsymbol{\pi}}) + \boldsymbol{\pi}^T \boldsymbol{D}^{(1)}.$$

By applying (6.55) iteratively we finally obtain

$$D(\theta \| \tilde{\theta}) \leq D(\boldsymbol{\pi} \| \tilde{\boldsymbol{\pi}}) + \boldsymbol{\pi}^T \left(\sum_{n=1}^{N-1} \mathsf{A}^{n-1} \boldsymbol{d} + \mathsf{A}^{N-1} \boldsymbol{e} \right). \tag{6.56}$$

Note that the KLD between two HMM's depends on the length of the observation sequence. To overcome this problem, by invoking ergodicity, Juang and Rabiner suggested the use of the following "normalized" KLD:

$$\bar{D}(\theta \| \tilde{\theta}) = \lim_{N \to \infty} \frac{1}{N} D(\theta \| \tilde{\theta}) \tag{6.57}$$

If we assume that the model θ is stationary, that is there exists a stationary distribution vector $\boldsymbol{s}$ such that $\boldsymbol{s}^T \mathsf{A} = \boldsymbol{s}^T$ and

$$\lim_{n \to \infty} \boldsymbol{\pi}^T \mathsf{A}^n = \boldsymbol{s}^T,$$

then by substituting (6.56) into (6.57) and taking the limit we obtain the following simple upper bound for the "normalized" KLD between two HMM's

$$\bar{D}(\theta \| \tilde{\theta}) \leq \boldsymbol{s}^T \boldsymbol{d} = \sum_{j=1}^{J} s_j \left(D(\boldsymbol{a}_j \| \tilde{\boldsymbol{a}}_j) + D(b_j \| \tilde{b}_j) \right). \tag{6.58}$$

Chapter 7

Conclusion

7.1 Summary

In the introduction, we explained the importance of having an efficient or *sparse representation* of visual information in many image processing tasks. The efficiency of a representation can be obtained and measured by a *non-linear approximation*, namely by keeping the M-best components in the expansion.

In the case of one-dimensional piecewise smooth signals, which is often encountered in practice (for example, one scan line of a typical image), wavelets provide the right transform. However, images are not simply stacks of one-dimensional piecewise smooth scan lines: discontinuity points or edges are typically positioned along geometrical smooth boundaries. As the result of a separable transform, two-dimensional wavelets impose some isotropy on the approximation process, and thus fail to capture the *geometrical regularity* presented in images. Still, the wavelets framework that allow signals to be successively approximated by localized basis elements is the foundation for building new systems.

A recent breakthrough in harmonic analysis came with the construction of the *ridgelet* and *curvelet* transforms by Candès and Donoho [21, 20]. Curvelets is a fixed transform that provides the optimal approximation rate for two-dimensional functions composed of piecewise smooth regions, and separated by smooth curves. The two key additional features of curvelets compared to separable wavelets are *directionality* and *anisotropy*. The curvelet transform was initially intended for functions defined on the *continuum* plane $\mathbb{R}^2$. Due to the directionality property, development of *discrete* versions of the curvelet transform that would lead to algorithmic implementations is a challenging problem. In particular, in certain applications, it is desirable to have invertible transform that is critically sampled or oversampled by a small factor.

Motivated by the success of *wavelets and filter banks*, we investigated a possibility to implement the discrete curvelet transform by using perfect reconstruction filter banks. The filter banks for curvelet are necessary non-separable and can provide a multiscale and directional decomposition for images. The

key is to use a *multiscale decomposition* to capture the *edge points* in the same way as wavelets, followed by a *directional decomposition* to link edge points into *contour segments*.

The *Laplacian pyramid* (LP) [18] is well-suited for the multiscale decomposition for the curvelet transform since it generates one subband image for each level that does not have the "frequency scrambling". Since the LP is an overcomplete representation it must be treated as a *frame* operator. For directional decomposition, we employed a *directional filter bank* originally proposed by Bamberger and Smith [9].

Inspired by the elegant *multiresolution analysis framework* that connects wavelet transform in harmonic analysis and filter banks in discrete signal processing, we investigated a similar connection between *curvelets and filter banks*. The key question is to obtain a correct joint sampling of *scale*, *space* and *direction*.

The ridgelet transform suggests a an alternative method to construct directional bases using the *Radon transform* followed by the wavelet transform. Our aim was to develop a *discrete ridgelet transform* that achieves both *invertibility* and *non-redundancy*.

Finally, we investigated the image retrieval application, in particular wavelet-based texture retrieval. The goal was to find a compact set of *features* that accurately represents the content of a texture image and obtain a good *similarity measure* between images using the feature sets.

The main contributions of the thesis are summarized below.

Framing Laplacian pyramid and new reconstruction

We proved that the Laplacian pyramid with orthogonal filters is a *tight frame* and proposed a new reconstruction method for the LP based on the *dual frame* operator. The new method presents an efficient filter bank, that leads to a proved improvement over the usual method for reconstruction in the presence of noise. Setting up the LP as an oversampled filter bank offers a parameterization of all synthesis filter banks that provides perfect reconstruction for the LP. This also leads to a connection between the iterated LP and a wavelet frame in the continuous domain.

Local directional bases from directional filter banks

We presented an in-depth study of the local directional bases generated by iterated directional filter banks (DFB). This was facilitated by an explicit formulas of equivalent filters and overall sampling matrices for each channel of the DFB. We examined in detail the "sinc" case, where the basis functions are assumed to have ideal frequency response. This leads to a connection between the iterated DFB and a *local Radon transform* in the continuous domain.

Pyramidal directional filter bank

We proposed a double filter bank structure, the *pyramidal directional filter bank* (PDFB), by combining the Laplacian pyramid with the directional filter bank, which provides a flexible multiscale and directional representation for images. The PDFB can be designed to be a tight frame that satisfies the anisotropy scaling relation for curves, and thus leads to an efficient implementation of the discrete curvelet transform for images. That originally designed to offer a sparse representation for two-dimensional piecewise smooth signals, the PDFB is also useful for image analysis and feature detection.

Directional multiresolution analysis

We defined a new *directional multiresolution analysis* where successive refinement happens at *both* spatial and directional resolution, using nested subspaces for each domain. This relates the constructed PDFB with the frames in the continuous domain and makes the connection with curvelets precise.

Orthonormal finite ridgelet transform

We developed a ridgelet implementation for finite-size images that is invertible, non-redundant and computed via fast algorithms. Furthermore, this construction leads to a new family of block directional bases for images, including adaptive schemes. Numerical results show that the new transform is more effective than the wavelet transform in approximating and denoising images with straight edges.

Statistical framework for image retrieval

We introduced a statistical framework for image retrieval by jointly considering the two problems of feature extraction and similarity measurement into a joint modeling and classification scheme, while taking into account the complexity constraint for such applications. The framework is asymptotically optimal in terms of retrieval error probability. Moreover, it can be applied to a general class of feature sets, leading to new similarity measurements and providing justification for existing methods

New methods for wavelet-based texture modeling and retrieval

We applied successfully the statistical framework in the wavelet-based texture retrieval application. The first method uses the generalized Gaussian density to model accurately the marginal distribution of wavelet coefficients. The second method uses the wavelet-domain hidden Markov model [38] to capture both the marginal distributions and the inter scales and orientations dependencies of wavelet coefficients, while being rotation invariant. Experimental results indicate that the new methods significantly improve retrieval rate over the tradi-

tional approach (which uses wavelet energy features and normalized Euclidean distance), while requiring comparable computational time.

7.2 Future Research

The development of new representation tools for images that can both capture geometrical regularity and be implemented by fast algorithms, is one of the most active research areas in the image processing community. The multi-dimensional filter bank approach in this thesis is very promising and leads to a variety of possible extensions. In the following, we give an overview of ongoing and future research directions.

Extending the Laplacian pyramid

In the standard structure of the LP, we can only design the pair of lowpass filters (H and G). The oversampled FB representation of the LP prompts us to extend it so that we have the freedom in designing all pairs of highpass filters (K_i and F_i), as well. This extension allows us to design the highpass filters with smaller supports while providing the same accuracy number and number of vanishing moments for the LP as in the standard structure. Moreover, the highpass filters can be designed to be (approximately) shifted versions of one filter so that the resulting difference signal in the LP is (nearly) shift invariant.

The analysis and design of oversampled FB's for wavelet frames is currently an area of active research [128, 43, 29, 146]. Several results can lead to 1-D LP with the above properties. However, they are specific for the 1-D case and can not be extended to the LP in higher dimensions by the separable method. Therefore, we have to look for new non-separable designs.

Regularity of the directional filter banks

Regularity for non-separable filter banks is a very challenging problem [96, 33]. For directional filter banks, the problem is even more complicated since iterations are carried out in all branches. We have shown that the "sinc" case leads to regularity functions in all directions. We expect that this is also possible with compactly supported filters.

New filter designs for the directional filter banks

Given the regularity conditions, the next step is to design filters that satisfy these criteria, as well as perfect reconstruction and vanishing moments. The experimental results presented in the thesis use long filters to simulate the "sinc" case, thus they are limited in spatial localization. We expect the performance of the PDFB to improve significantly once better filters are found.

Multiscale and directional bases

Since the PDFB is a redundant transform, an obvious question is can we modify it to be a basis? The redundancy of the PDFB is due to the LP and can be fixed, for example, by replacing the LP with a quincunx filter bank for the multiscale decomposition. However, this approach creates "frequency scrambling" where frequency is not correctly partitioned and thus could be problematic in the applications. Nevertheless, it is still worthwhile to investigate the property of resulting bases. An alternative approach is to search for a different multiscale and directional sampling scheme.

Exploring tree data structures for image coding

The success of a representation in coding application is not just depending on good approximation properties but also on the existence of effective data structure for indexing significant coefficients [32]. The good performance of wavelets in the state-of-the-art image compression algorithms relies on the use of embedded trees that effectively point to (in)significant coefficients in the transform domain [147, 143]. The PDFB has a pyramid structure so that resolution increases in both spatial and direction domain when going from coarse to fine scale. Moreover, as smooth edges have their tangents varying slowly, significant coefficients in PDFB are also localized in both position and direction. Therefore, tree data structures for the PDFB coefficients that allow algorithms successively locate the position *and* direction of image edges seem to be very promising in coding and other applications.

Bibliography

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Tables*. Dover Publications, New York, 1970.

[2] A. Aldroubi. Portraits of frames. *Proc. Amer. Math. Soc.*, 123:1661–1668, 1995.

[3] R. Ansari and C. Guillemot. Exact reconstruction filter banks using diamond FIR filters. In *Proc. Bilkent Intl. Conf.*, pages 1412–1424, July 1990.

[4] J. P. Antoine, P. Carrette, R. Murenzi, and B. Piette. Image analysis with two-dimensional continuous wavelet transform. *Signal Processing*, 31:241–272, 1993.

[5] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Trans. Image Proc.*, 1(2):205–220, April 1992.

[6] A. Averbuch, R. R. Coifman, D. L. Donoho, M. Israeli, and J. Waldén. Fast slant stack: A notion of Radon transform for data in a Cartesian grid which is rapidly computable, algebraically exact, geometrically faithful and invertible. Technical report, 2001. Preprint, `http://www.math.tau.ac.il/~amir1/`.

[7] R. H. Bamberger. *The Directional Filter Bank: A Multirate Filter Bank for the Directional Decomposition of Images*. PhD thesis, Georgia Institute of Technology, 1990.

[8] R. H. Bamberger. New results on two and three dimensional directional filter banks. In *Twenty-Seventh Asilomar Conf. on Signals, Systems and Computers*, volume 2, pages 1286–1290, 1993.

[9] R. H. Bamberger and M. J. T. Smith. A filter bank for the directional decomposition of images: Theory and design. *IEEE Trans. Signal Proc.*, 40(4):882–893, April 1992.

[10] J. Benedetto and S. D. Li. The theory of multiresolution analysis frames and applications to filter banks. *Journal of Appl. and Comput. Harmonic Analysis*, 5(4):389–427, 1998.

[11] J. R. Bergen and E. H. Adelson. Theories of visual texture perception. In D. Regan, editor, *Spatial Vision*. CRC press, 1991.

[12] G. Beylkin. Discrete Radon transform. *IEEE Trans. Acoust. Speech Sig. Proc.*, 35:162–172, 1987.

[13] H. Bölcskei and F. Hlawatsch. Noise reduction in oversampled filter banks using predictive quantization. *IEEE Trans. Inform. Th.*, 47(1):155–172, Jan. 2001.

[14] H. Bölcskei, F. Hlawatsch, and H. G. Feichtinger. Frame-theoretic analysis of oversampled filter banks. *IEEE Trans. Signal Proc.*, 46:3256–3268, Dec 1998.

[15] E. D. Bolker. The finite Radon transform. In S. Helgason R. L. Bryant, V. Guillemin and R. O. Wells Jr., editors, *Integral Geometry (Contemporary Mathematics, Vol. 63)*, pages 27–50. 1987.

[16] J. S. De Bonet and P. Viola. Texture recognition using a non-parametric multi-scale statistical model. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1998.

[17] C. M. Brislawn. Classification of nonexpansive symmetric extension transforms for multirate filter banks. *Applied and Computational Harmonic Analysis*, 3:337–357, 1996.

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

[19] E. J. Candès. *Ridgelets: Theory and Applications*. PhD thesis, Department of Statistics, Stanford University, 1998.

[20] E. J. Candès and D. L. Donoho. Curvelets – a suprisingly effective nonadaptive representation for objects with edges. In A. Cohen, C. Rabut, and L. L. Schumaker, editors, *Curve and Surface Fitting*, Saint-Malo, 1999. Vanderbilt University Press.

[21] E. J. Candès and D. L. Donoho. Ridgelets: a key to higher-dimensional intermittency? *Phil. Trans. R. Soc. Lond. A.*, pages 2495–2509, 1999.

[22] E. J. Candès and D. L. Donoho. Curvelets, multiresolution representation, and scaling laws. In A. Aldroubi, A. F. Laine, and M. A. Unser, editors, *SPIE Wavelet Applications in Signal and Image Processing VIII*, volume 4119, 2000.

[23] J. W. Cassels. *An Introduction to the Geometry of Numbers*. Springer-Verlag, Berlin, 1971.

[24] T. Chang and C.-C. J. Kuo. Texture analysis and classification with tree-structure wavelet transform. *IEEE Trans. Image Proc.*, 2(4):429–441, 1993.

[25] J.-L. Chen and A. Kundu. Rotational and gray-scale transform invariant texture identification using wavelet decomposition and hidden Markov model. *IEEE Trans. Patt. Recog. and Mach. Intell.*, 16:208–214, Feb. 1994.

[26] J. Y. Chen, C. A. Bouman, and J. P. Allebach. Multiscale branch and bound algorithm image database. In *Proc. of SPIE/IS&T Conf. on Storage and Retrieval for Image and Video Data bases V*, volume 3022, pages 133–144, San Jose CA, 1997.

[27] T. Chen and P. P. Vaidyanathan. Recent developments in multidimensional multirate systems. *IEEE Trans. on CSVT*, 3(2):116–137, April 1993.

[28] H. Choi and R. G. Baraniuk. Image segmentation using wavelet-domain classification. In *Proc. of SPIE Conf. on Mathematical Modeling, Bayesian Estimation, and Inverse Problems*, volume 3816, pages 306–320, 1999.

[29] C. K. Chui, W. He, and J. Stöckler. Compactly supported tight and sibling frames with maximum vanishing moments. Technical Report 2001-04, Department of Statistics, Stanford University, 2001. `http://www-stat.stanford.edu/research/`.

[30] A. Cohen and I. Daubechies. Nonseparable bidimensional wavelet bases. *Rev. Mat. Iberoamericana*, 9(1):51–137, 1993.

[31] A. Cohen, I. Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Commun. on Pure and Appl. Math.*, 45:485–560, 1992.

[32] A. Cohen, I. Daubechies, O. G. Guleryuz, and M. T. Orchard. On the importance of combining wavelet-based nonlinear approximation with coding strategies. *IEEE Trans. Info. Theory*, 2000. submitted.

[33] A. Cohen, K. Gröchenig, and L. F. Villemoes. Regularity of multivariate refinable functions. *Constr. Approx.*, 15:241–255, 1999.

[34] R. R. Coifman, Y. Meyer, and M. V. Wickerhauser. Wavelet analysis and signal processing. In M. B. Ruskai et al, editor, *Wavelets and their Applications*, pages 153–178. Jones and Barlett, Boston, 1992.

[35] R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory, Special Issue on Wavelet Transforms and Multiresolution Signal Analysis*, 38(2):713–718, March 1992.

[36] JPEG committee. JPEG home page. `http://www.jpeg.org`.

[37] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley Interscience, New York, NY, 1991.

[38] M. Crouse, R. D. Nowak, and R. G. Baraniuk. Wavelet-based signal processing using hidden Markov models. *IEEE Trans. Signal Proc. (Special Issue on Wavelets and Filterbanks)*, pages 886–902, Apr. 1998.

[39] Z. Cvetković and M. Vetterli. Oversampled filter banks. *IEEE Trans. Signal Proc.*, 46:1245 –1255, May 1998.

[40] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Commun. on Pure and Appl. Math.*, 41:909–996, November 1988.

[41] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Trans. Inform. Th.*, 36(5):961–1005, September 1990.

[42] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, PA, 1992.

[43] I. Daubechies, B. Han, A. Ron, and Z. Shen. Framelets: MRA-based constructions of wavelet frames. Preprint, `ftp://ftp.cs.wisc.edu/Approx/dhrs.ps`, 2001.

[44] J. Daugman. Two-dimensional spectral analysis of cortical receptive field profile. *Vision Research*, 20:847–856, 1980.

[45] S. R. Deans. *The Radon Transform and Some of Its Applications*. John Wiley & Sons, 1983.

[46] R. A. DeVore. Nonlinear approximation. *Acta Numer.*, 7:51–150, 1998.

[47] R. A. DeVore, B. Jawerth, and B. J. Lucier. Image compression through wavelet transform coding. *IEEE Transactions on Information Theory, Special Issue on Wavelet Transforms and Multiresolution Signal Analysis*, 38(2):719–746, March 1992.

[48] M. Do, S. Ayer, and M. Vetterli. Invariant image retrieval using wavelet maxima moment. In *Proc. of 3rd Int. Conf. in Visual Information and Information Systems*, pages 451–458, 1999.

[49] M. N. Do. Fast approximation of Kullback-Leibler distance for dependence trees and hidden Markov models. *IEEE Signal Proc. Letters*, Oct. 2001. submitted.

[50] M. N. Do, A. C. Lozano, and M. Vetterli. Rotation invariant texture retrieval using steerable wavelet-domain hidden Markov models. In *Proc. of SPIE Conf. on Wavelet Applications in Signal and Image Processing VIII*, San Diego, USA, Aug. 2000.

[51] M. N. Do and M. Vetterli. Image denoising using orthonormal finite ridgelet transform. In *Proc. of SPIE Conf. on Wavelet Applications in Signal and Image Processing VIII*, San Diego, USA, Aug. 2000.

[52] M. N. Do and M. Vetterli. Orthonormal finite ridgelet transform for image compression. In *Proc. IEEE Int. Conf. on Image Proc.*, Vancouver, Canada, Sep. 2000.

[53] M. N. Do and M. Vetterli. Texture similarity measurement using Kullback-Leibler distance on wavelet subbands. In *Proc. IEEE Int. Conf. on Image Proc.*, Vancouver, Canada, Sep. 2000.

[54] M. N. Do and M. Vetterli. Curvelets and filter banks. *IEEE Trans. Image Proc.*, 2001. to be submitted.

[55] M. N. Do and M. Vetterli. The finite ridgelet transform for image representation. *IEEE Trans. Image Proc.*, 2001. submitted.

[56] M. N. Do and M. Vetterli. Frame reconstruction of the Laplacian pyramid. In *IEEE Trans. Acoust., Speech, and Signal Proc.*, Salt Lake City, May 2001.

[57] M. N. Do and M. Vetterli. Framing pyramids. *IEEE Trans. Signal Proc.*, Oct. 2001. submitted.

[58] M. N. Do and M. Vetterli. Pyramidal directional filter banks and curvelets. In *Proc. IEEE Int. Conf. on Image Proc.*, Thessaloniki, Greece, Oct. 2001.

[59] M. N. Do and M. Vetterli. Rotation invariant texture characterization and retrieval using steerable wavelet-domain hidden Markov models. *IEEE Trans. Multimedia*, Apr. 2001. submitted.

[60] M. N. Do and M. Vetterli. Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance. *IEEE Trans. Image Proc.*, 2001. to appear.

[61] D. Donoho and I. Johnstone. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, pages 425–455, December 1994.

[62] D. L. Donoho. Sparse component analysis and optimal atomic decomposition. *Constructive Approximation*, 17:353–382, 1998.

[63] D. L. Donoho. Wedgelets: nearly-minimax estimation of edges. *Ann. Statist.*, 27:859–897, 1999.

[64] D. L. Donoho and M. R. Duncan. Digital curvelet transform: strategy, implementation and experiments. In *Proc. Aerosense 2000, Wavelet Applications VII*, volume 4056, pages 12–29. SPIE, 2000. `http://www-stat.stanford.edu/~donoho/Reports/1999/DCvT.pdf`.

[65] D. L. Donoho, M. Vetterli, R. A. DeVore, and I. Daubechies. Data compression and harmonic analysis. *IEEE Trans. Inform. Th.*, 44(6):2435–2476, October 1998.

[66] P. L. Dragotti and M. Vetterli. Footprints and edgeprints for image denoising and compression. In *Proc. IEEE Int. Conf. on Image Proc.*, Thessaloniki, Greece, Oct. 2001.

[67] E. Dubois. The sampling and reconstruction of time-varying imagery with application in video systems. *Proc. IEEE*, 73(4):502–522, April 1985.

[68] R. J. Duffin and A. C. Schaeffer. A class of nonharmonic Fourier series. *Trans. Amer. Math. Soc.*, 72:341–366, 1952.

[69] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Stelle, and P. Yanker. Query by image and video content: The QBIC system. *Computer*, pages 23–32, September 1995.

[70] D. Gabor. Guest editorial. *IRE Trans. Info. Theory*, Sep. 1959.

[71] I. Gertner. A new efficient algorithm to compute the two-dimensional discrete Fourier transform. *IEEE Trans. Acoust. Speech Sig. Proc.*, 36:1036–1050, Jul 1988.

[72] V. K. Goyal, J. Kovačević, and J. A. Kelner. Quantized frame expansions with erasures. *Applied and Computational Harmonic Analysis*, 10(3):203–233, May 2001.

[73] V. K Goyal, M. Vetterli, and N. T. Thao. Quantized overcomplete expansions in $\mathbb{R}^n$: Analysis, synthesis and algorithms. *IEEE Trans. Inform. Th.*, 44(1):16–31, Jan. 1998.

[74] H. Greenspan, S. Belongie, R. Goodman, and P. Perona. Rotation invariant texture recognition using a steerable pyramid. In *Proc. Int. Conf. on Pattern Recognition*, pages 162–167, 1994.

[75] R. P. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction*. Addison-Wesley, 3rd edition, 1994.

[76] G. M. Haley and B. S. Manjunath. Rotation-invariant texture classification using a complete space-frequency model. *IEEE Trans. Image Proc.*, 8:255–269, Feb. 1999.

[77] D. Heeger and J. Bergen. Pyramid-based texture analysis/synthesis. In *Proc. ACM SIGGRAPH*, 1995.

[78] C. Herley and M. Vetterli. Wavelets and recursive filter banks. *IEEE Trans. Signal Proc.*, 41(8):2536–2556, August 1993.

[79] G. T. Herman. *Image Reconstruction from Projections: The Fundamentals of Computerized Tomography*. Academic Press, 1980.

[80] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[81] T. S. Huang, S. Mehrotra, and K. Ramchandran. Multimedia analysis and retrieval system (MARS) project. In *Proc. of 33rd Annual Clinic on Library Application on Data Processing - Digital Image Access and Retrieval*, 1996.

[82] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, (160):106–154, 1962.

[83] C.E. Jacobs, A. Finkelstein, and D.H. Salesin. Fast multiresolution image querying. In *Computer graphics proceeding of SIGGRAPH*, pages 278–280, Los Angeles, 1995.

[84] A.K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc, 1989.

[85] B. H. Juang and L. R. Rabiner. A probabilistic distance measure for hidden Markov models. *AT & T Tech. J.*, 64(2):391–408, Feb. 1985.

[86] A. Karasaridis and E. Simoncelli. A filter design technique for steerable pyramid image transforms. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, Atlanta, May 1996.

[87] G. Karlsson and M. Vetterli. Theory of two - dimensional multirate filter banks. *IEEE Trans. Acoust., Speech, and Signal Proc.*, 38(6):925–937, June 1990.

[88] S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, 1993.

[89] B. T. Kelley and V. K. Madisetti. The fast discrete Radon transform – I: Theory. *IEEE Trans. on Image Proc.*, 2:382–400, Jul 1993.

[90] N. Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Journal of Appl. and Comput. Harmonic Analysis*, 10:234–253, 2001.

[91] J. Kovačević. *Filter Banks and Wavelets: Extensions and Applications*. PhD thesis, Columbia University, Oct. 1991.

[92] J. Kovačević, P. L. Dragotti, and V. K. Goyal. Filter bank frame expansions with erasures. *IEEE Trans. Inform. Th.*, 2001. submitted.

[93] J. Kovačević and M. Vetterli. Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for $\mathcal{R}^n$. *IEEE Transactions on Information Theory, Special Issue on Wavelet Transforms and Multiresolution Signal Analysis*, 38(2):533–555, March 1992.

[94] J. Kovačević and M. Vetterli. Nonseparable two- and three-dimensional wavelets. *IEEE Trans. Signal Proc.*, 43(5):1269–1273, May 1995.

[95] A. Laine and J. Fan. Texture classification by wavelet packet signatures. *IEEE Trans. Patt. Recog. and Mach. Intell.*, 15:1186–1191, 1993.

[96] W. M. Lawton and H. L. Resnikoff. Multidimensional wavelet bases. Aware Report AD910130, Aware, Inc., Cambridge, MA-02138, 1991.

[97] J. Lebrun and I. Selesnick. Gröbner bases and wavelet design. *Journal of Symbolic Computation*, 2001. to appear.

[98] R. Lidl and H. Niederreiter. *Introduction to Finite Fields and Their Applications.* Cambridge University Press, revised edition, 1994.

[99] Y.-P. Lin and P. P. Vaidyanathan. Theory and design of two-dimensional filter banks: A review. *Multidimensional Systems and Signal Proc.*, 7:263–330, 1996.

[100] F. Liu and R. W. Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE Trans. Patt. Recog. and Mach. Intell.*, 18(7):722–733, July 1996.

[101] S. Mallat. Multiresolution approximations and wavelet orthonormal bases of $L_2(R)$. *Trans. Amer. Math. Soc.*, 315:69–87, September 1989.

[102] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Patt. Recog. and Mach. Intell.*, 11(7):674–693, July 1989.

[103] S. Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, 2nd edition, 1999.

[104] S. Mallat and W. L. Hwang. Singularity detection and processing with wavelets. *IEEE Transactions on Information Theory, Special Issue on Wavelet Transforms and Multiresolution Signal Analysis*, 38(2):617–643, March 1992.

[105] S. Mallat and S. Zhong. Wavelet maxima representation. In Y. Meyer, editor, *Wavelets and Applications*, pages 207–284. Masson, Paris, 1991.

[106] H. S. Malvar. *Signal Processing with Lapped Transforms.* Artech House, Norwood, MA, 1992.

[107] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Patt. Recog. and Mach. Intell.*, 18(8):837–842, August 1996.

[108] F. Matúš and J. Flusser. Image representation via a finite Radon transform. *IEEE Trans. Pattern Anal. Machine Intell.*, 15(10):996–1006, Oct 1993.

[109] J. McClellan. The design of two-dimensional filters by transformations. In *Seventh Ann. Princeton Conf. on ISS*, pages 247–251, Princeton, NJ, 1973.

[110] R. M. Mersereau and A. V. Oppenheim. Digital reconstruction of multidimensional signals from projections. *Proc. of IEEE*, 62(10):1319–1338, Oct 1974.

[111] F. G. Meyer and R. R. Coifman. Brushlets: a tool for directional image analysis and image compression. *Journal of Appl. and Comput. Harmonic Analysis*, 5:147–187, 1997.

[112] Y. Meyer. *Wavelets and Operators*. Advanced mathematics. Cambridge University Press, 1992.

[113] MIT Vision and Modeling Group. Vision texture. `http://vismod.www.media.mit.edu`.

[114] A. Mojsilovic, J. Kovacevic, J. Hu, J. Safranek, and S. K. Ganapathy. Matching and retrieval based on the vocabulary and grammar of color patterns. *IEEE Trans. Image Proc.*, 9(1):38–54, Jan. 2000.

[115] P. Moulin and J. Liu. Analysis of multiresolution image denoising schemes using generalized Gaussian and complexity priors. *IEEE Trans. Inform. Th.*, 45:909–919, 1999.

[116] I. Newton. New theory about light and colors. *Philosophical Transactions of the Royal Society*, 7(80):3075–3087, 1672.

[117] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, (381):607–609, 1996.

[118] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1989.

[119] A. Papoulis. *Probability, Random Variables and Stochastic Processes, Second Edition*. McGraw-Hill, New York, NY, 1984.

[120] H. Park, T. Kalker, and M. Vetterli. Gröbner bases and multidimensional fir multirate systems. *Multidimensional Systems and Signal Processing, Kluwer Acad. Pub*, 8:11–30, 1997.

[121] S. Park. *New Directional Filter Banks and Their Applications in Image Processing*. PhD thesis, Georgia Institute of Technology, 1999.

[122] S. Park, M. J. T. Smith, and R. M. Mersereau. A new directional filterbank for image analysis and classification. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, pages 1417–1420, 1999.

[123] Z. Pecenovic, M. Do, S. Ayer, and M. Vetterli. New methods for image retrieval. In *Proceedings of ICPS'98 Congress on Exploring New Tracks in Imaging*, September 1998.

[124] Z. Pecenovic, M. N. Do, M. Vetterli, and P. Pu. Integrated browsing and searching of large image collections. In *Proc. of 4th International Conf. on Visual Information Systems*, Lyon, France, Nov. 2000.

[125] E. Le Pennec and S. Mallat. Image compression with geometric wavelets. In *Proc. IEEE Int. Conf. on Image Proc.*, Vancouver, Canada, Sep. 2000.

[126] A. Pentland, R.W. Piccard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233–254, 1996.

[127] A. Petukhov. Explicit construction of framelets. Technical Report 00:03, Industrial Mathematics Institute, University of South Carolina, 2000. `http://www.math.sc.edu/~imip/`.

[128] A. Petukhov. Symmetric framelets. Technical Report 00:15, Industrial Mathematics Institute, University of South Carolina, 2000. `http://www.math.sc.edu/~imip/`.

[129] 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 Proc.*, 43(3):649–665, Mar. 1995.

[130] H. V. Poor. *An Introduction to Signal Estimation and Detection*. Springer-Verlag, New York, 2nd edition, 1994.

[131] B. Porat. *Digital Processing of Random Signals: Theory and Methods*. Prentice-Hall, Englewood Cliffs, NJ, 1994.

[132] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, Feb. 1989.

[133] K. Ramchandran, A. Ortega, and M. Vetterli. Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders. *IEEE Trans. Image Proc.*, 3(5):533–545, September 1994.

[134] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Trans. Image Proc.*, 2(2):160–175, April 1993.

[135] T. Randen and J. H. Husoy. Filtering for texture classification: a comparative study. *IEEE Trans. Patt. Recog. and Mach. Intell.*, 21:291–310, 1999.

[136] A. R. Rao and G. L. Lohse. Towards a texture naming system: Identifying relevant dimensions of texture. *Visual Res.*, 36(11):1649–1669, 1996.

[137] K. R. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, 1990.

[138] J. K. Romberg, H. Choi, and R. G. Baraniuk. Bayesian tree-structured image modeling using wavelet-domain hidden Markov models. In *Proc. of SPIE Conf. on Mathematical Modeling, Bayesian Estimation, and Inverse Problems*, volume 3816, pages 306–320, 1999.

[139] A. Ron and Z. Shen. Affine systems in $L_2(\mathbb{R}^d)$: The analysis of the analysis operator. *J. Functional Anal.*, (148):408–447, 1997.

[140] O. Ronen, J. R. Rohlicek, and M. Ostendorf. Parameter estimation of dependence tree models using the EM algorithm. *IEEE Signal Proc. Letters*, 2(8):157–159, Aug. 1995.

[141] A. Rosenfeld, editor. *Multiresolution Image Processing and Analysis*. Springer-Verlag, New York, 1984.

[142] A. Rosenfeld and A. C. Kak. *Digital Picture Processing*. Academic Press, 2nd edition, 1982.

[143] A. Said and W. A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits and Systems for Video Technology*, 6(3):243–250, Jun. 1996.

[144] P. M. Salzberg and R. Figueroa. Tomography on the 3D-torus and crystals. In G. T. Herman and A. Kuba, editors, *Discrete Tomography: Foundations, Algorithms and Applications*, pages 417–434. Birkhäuser, 1999.

[145] J. L. C. Sanz, E. B. Hinkle, and A. K. Jain. *Radon and Projection Transform-Based Computer Vision*. Springer Verlag, 1988.

[146] I. W. Selesnick. The double density DWT. In A. Petrosian and F. G. Meyer, editors, *Wavelets in signal and image analysis: from theory to practice*. Kluwer, 2001. `http://taco.poly.edu/selesi/`.

[147] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing, Special Issue on Wavelets and Signal Processing*, 41(12):3445–3462, December 1993.

[148] K. Sharifi and A. Leon-Garcia. Estimation of shape parameter for generalized Gaussian distributions in subband decompositions of video. *IEEE Trans. Circuits Sys. Video Tech.*, 5:52–56, 1995.

[149] Y. Shoham and A. Gersho. Efficient bit allocation for an arbitrary set of quantizers. *IEEE Trans. Acoust., Speech, and Signal Proc.*, 36(9):1445–1453, September 1988.

[150] E. P. Simoncelli and E. H. Adelson. Subband transforms. In J. W. Woods, editor, *Subband Image Coding*, pages 143–192. Kluwer Academic Publishers, Inc., Boston, MA, 1991.

[151] E. P. Simoncelli and W. T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *Proc. of IEEE Int'l Conf. on Image Processing*, Washington DC, 1995.

[152] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger. Shiftable multiscale transforms. *IEEE Transactions on Information Theory, Special Issue on Wavelet Transforms and Multiresolution Signal Analysis*, 38(2):587–607, March 1992.

[153] E. P. Simoncelli and J. Portilla. Texture characterization via joint statistics of wavelet coefficient magnitudes. In *Proc. of IEEE Int'l Conf. on Image Processing*, 1998.

[154] Y. Singer and M. K. Warmuth. Batch and on-line parameter estimation of Gaussian mixtures based on the joint entropy. *Advances in Neural Information Processing Systems 11 (NIPS'98)*, pages 578–584, 1998.

[155] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Patt. Recog. and Mach. Intell.*, 22:1349–1380, Dec. 2000.

[156] J. R. Smith and S.-F. Chang. Transform features for texture classification and discrimination in large image databases. In *Proc. of IEEE Int. Conf. on Image Processing*, 1994.

[157] J.R. Smith and S.-F. Chang. VisualSEEk: a fully automated content-based image query system. In *Proc. The Fourth ACM International Multimedia Conference*, pages 87–98, November 1996.

[158] D. McG. Squire and T. Pun. Assessing agreement between human and machine clusterings of image databases. *Pattern Recognition*, 31(12):1905–1919, 1998.

[159] J. L. Starck, E. J. Candès, and D.L. Donoho. The curvelet transform for image denoising. *IEEE Trans. Image Proc.*, 2000. submitted.

[160] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley Cambridge Press, Boston, 1996.

[161] M. Swain and D. Ballard. Color indexing. *Int. Journal of Computer Vision*, vol. 7(1), 1991.

[162] H. Tamura, S. Mori, and T. Yamawaki. Texture features corresponding to visual perception. *IEEE Trans. Systems, Man and Cybernetics*, 8:460–473, 1982.

[163] C. W. Therrien. *Decision, Estimation, and Classification: an Introduction to Pattern Recognition and Related Topics*. John Wiley & Sons, 1989.

[164] University of Southern California, Signal and Image Processing Institute. Rotated textures. `http://sipi.usc.edu/services/database/Database.html`.

[165] M. Unser. Local linear transforms for texture measurements. *Signal Processing*, 11:61–79, 1986.

[166] M. Unser. An improved least squares Laplacian pyramid for image compression. *Signal Proc.*, 27:187–203, May 1992.

[167] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Trans. Image Proc.*, 4:1549–1560, 1995.

[168] M. Unser and A. Aldroubi. A general sampling theory for non-ideal acquisition devices. *IEEE Trans. Signal Proc.*, pages 2915–2925, 1994.

[169] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice-Hall, Englewood Cliffs, NJ, 1993.

[170] M. K. Varanasi and B. Aazhang. Parametric generalized Gaussian density estimation. *J. Acoust. Soc. Amer.*, 86:1404–1415, 1989.

[171] N. Vasconcelos and A. Lippman. A unifying view of image similarity. In *Proc. of IEEE Int. Conf. on Pattern Recognition (ICPR)*, Barcelona, Spain, 2000.

[172] M. Vetterli. Multidimensional subband coding: Some theory and algorithms. *Signal Proc.*, 6(2):97–112, February 1984.

[173] M. Vetterli. Wavelets, approximation and compression. *IEEE SP Mag.*, pages 59–73, Sep. 2001.

[174] M. Vetterli and C. Herley. Wavelets and filter banks: Theory and design. *IEEE Trans. Signal Proc.*, 40(9):2207–2232, September 1992.

[175] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding*. Prentice-Hall, Englewood Cliffs, NJ, 1995.

[176] E. Viscito and J. P. Allebach. The analysis and design of multidimensional FIR perfect reconstruction filter banks for arbitrary sampling lattices. *IEEE Trans. Circ. and Syst.*, 38(1):29–42, January 1991.

[177] N. D. Vvedenskaya and S. G. Gindikin. Discrete Radon transform and image reconstruction. In I. M. Gelfand and S. G. Gindikin, editors, *Mathematical problems of tomography (Translations of mathematical monographs, Vol. 81)*, pages 141–188. American Mathematical Society, 1990.

[178] A. B. Watson. The cortex transform: Rapid computation of simulated neural images. *Computer Vision, Graphics, and Image Processing*, 39(3):311–327, 1987.

[179] G. V. Wouwer, P. Scheunders, and D. V. Dyck. Statistical texture characterization from discrete wavelet representations. *IEEE Trans. Image Proc.*, 8(4):592–598, April 1999.

[180] W.-R. Wu and S.-C. Wei. Rotation and gray-scale transform-invariant texture classification using spiral resampling, subband demposition, and hidden Markov model. *IEEE Trans. Image Proc.*, 5:1423–1434, Oct. 1996.

[181] S. C. Zhu, Y. N. Wu, and D. Mumford. FRAME: Filters, random field and maximum entropy: – towards a unified theory for texture modeling. *Int'l Journal of Computer Vision*, 27(2):1–20, 1998.

Curriculum Vitae

Minh N. Do
Audio-Visual Communications Laboratory
Swiss Federal Institute of Technology Lausanne (EPFL)
1015 Lausanne, Switzerland
Email: minh.do@ieee.org
Web: http://lcavwww.epfl.ch/~minhdo

Education

1998 - 2001 Ph.D. candidate in Department of Communication Systems, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland.

1997 - 1998 Graduate School in Communication Systems, EPFL.

1994 - 1997 Bachelor of Engineering in Computer Engineering (First Class Honors), University of Canberra, Australia.

Professional Experience

1998 - 2001 **Research and Teaching Assistant**, Audio-Visual Communications Laboratory, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland.

1996 - 1997 **Research Assistant** in Artificial Intelligence, Faculty of Information Sciences and Engineering, University of Canberra, Australia.

Summers 1995 & 1996 **Summer Research Scholar** in Lossless Image Compression, School of Computer Science, Australian Defence Force Academy, Canberra, Australia.

1995 - 1997 **Part-time Tutor**, Faculty of Information Sciences and Engineering, University of Canberra, Australia.

Awards and Honors

- Swiss Federal Institute of Technology Lausanne three-year Ph.D. fellowship awarded to the two best students from the Graduate School in Communication Systems (1998 - 2001).
- University of Canberra Medal for outstanding academic achievement (1997).
- First Prize of the Institution of Radio and Electronics Engineers Society (Canberra branch) in the Annual Final Year Engineering Project Presentation (1997).
- AusAID scholarship to outstanding first year students at the universities in Vietnam to carry on undergraduate study in Australia (1993 - 1997).
- Second Prize in the 32nd International Mathematical Olympiad, Sweden (1991).

Personal

Date of birth:	September 17, 1974.
Nationality:	Vietnam.
Civil status:	Married with one child.

Publications

Journals and Manuscripts

1. M. N. Do and M. Vetterli. Curvelets and filter banks. To be submitted, *IEEE Transactions on Image Processing*.
2. M. N. Do and M. Vetterli. Iterated directional filter banks: Construction and regularity. To be submitted, *IEEE Transactions on Signal Processing*.
3. M. N. Do and M. Vetterli. Framing pyramids. Submitted, *IEEE Transactions on Signal Processing*, December 2001.
4. M. N. Do. Fast approximating Kullback-Leibler distance for dependence tree and hidden Markov models. Submitted, *IEEE Signal Processing Letters*, December 2001.
5. M. N. Do and M. Vetterli. Rotation invariant texture characterization and retrieval using steerable wavelet-domain hidden Markov models. Submitted, *IEEE Transactions on Multimedia*, April 2001.
6. M. N. Do and M. Vetterli. The finite ridgelet transform for image representation. Accepted, *IEEE Transactions on Image Processing*, 2001.

7. M. N. Do and M. Vetterli. Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance. *IEEE Transactions on Image Processing*, February 2002, to appear.

Conferences

1. M. N. Do, P. L. Dragotti, R. Shukla and M. Vetterli. On compression of two-dimensional piecewise smooth functions. Invited paper, *IEEE International Conference on Image Processing, Special Session on Image Processing and Non-Linear Approximation*, Thessaloniki, Greece, October 2001.

2. M. N. Do and M. Vetterli. Pyramidal directional filter banks and curvelets. *IEEE International Conference on Image Processing*, Thessaloniki, Greece, October 2001.

3. M. N. Do and M. Vetterli. Frame reconstruction of the Laplacian pyramid. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, 2001.

4. M. N. Do and M. Vetterli. New non-separable transform with application to image compression. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, USA, May 2001.

5. M. N. Do and M. Vetterli. Texture characterization and retrieval using steerable hidden Markov models. *MSRI Workshop on Nonlinear Estimation and Classification*, Berkeley, USA, March 2001.

6. R. Shukla, M. N. Do and M. Vetterli. Best adaptive tiling in a rate-distortion sense. *MSRI Workshop on Nonlinear Estimation and Classification*, Berkeley, USA, March 2001.

7. Z. Pecenovic, M. N. Do, M. Vetterli and P. Pu. Integrated browsing and searching of large image collections. In *Proc. of 4th International Conference on Visual Information Systems*, pp. 279-289, Lyon, France, November 2000.

8. M. N. Do and M. Vetterli. Orthonormal finite ridgelet transform for image compression. In *Proc. of IEEE International Conference on Image Processing*, Vol. II, pp. 367-370, Vancouver, Canada, September 2000.

9. M. N. Do and M. Vetterli. Texture similarity measurement using Kullback-Leibler distance on wavelet subbands. In *Proc. of IEEE International Conference on Image Processing*, Vol. III, pp. 730-733, Vancouver, Canada, September 2000.

10. M. N. Do and M. Vetterli. Image denoising using orthonormal finite ridgelet transform. In *Proc. of SPIE Conference on Wavelet Applications in Signal and Image Processing*, San Diego, USA, August 2000.

11. M. N. Do, A. C. Lozano and M. Vetterli. Rotation invariant texture retrieval using steerable wavelet-domain hidden Markov models. In *Proc. of SPIE Conference on Wavelet Applications in Signal and Image Processing*, San Diego, USA, August 2000.

12. M. N. Do and M. Vetterli. *Orthonormal finite ridgelet transform for image denoising and compression*, NSF-CBMS Conference on Interactions of Harmonic Analysis, Statistical Estimation and Data Compression, University of Missouri - St. Louis, USA, May 2000.

13. M. N. Do and M. Vetterli. Wavelet-based texture characterization with application to content-based image retrieval. *Wavelets, Harmonic Analysis and Image Processing Workshop*, Technische Universität München, Germany, December 1999.

14. M. Do, S. Ayer and M. Vetterli. Invariant image retrieval using wavelet maxima moment. In *Proc. of 3rd International Conference on Visual Information and Information Systems*, pp. 451-458, Amsterdam, The Netherlands, June 1999.

15. Z. Pecenovic, M. Do, S. Ayer and M. Vetterli. New methods for image retrieval. In *Proc. of ICPS'98 Congress on Exploring New Tracks in Imaging*, pp. 242-246, Antwerp, Belgium, September 1998.

16. M. Do, S. Ayer and M. Vetterli. Invariant image retrieval using wavelet maxima moment. *Wavelet and Applications Workshop - Swiss Science Foundation*, Ascona, Switzerland, October 1998.

17. D. Tran, M. Do, M. Wagner and T.V. Le. A proposed decision rule for speaker identification based on a posteriori probability. In *Proc. of Workshop on Speaker Recognition and Its Commercial and Forensic Applications*, pp. 85-88, Avignon, France, April 1998.

18. M. Do and M. Wagner. Speaker recognition with small training requirements using a combination of VQ and DHMM. In *Proc. of Workshop on Speaker Recognition and Its Commercial and Forensic Applications*, pp. 169-172, Avignon, France, April 1998.