Optimally Localized Wavelets and Smoothing Kernels

Kunal Narayan Chaudhury

Thèse N° 4968 (February 2011)

Thèse présentée à la faculté des sciences et techniques de l'ingénieur
pour l'obtention du grade de docteur ès sciences
et acceptée sur proposition du jury

Prof. Pierre Vandergheynst, *président*
Prof. Michael Unser, *directeur de thèse*
Prof. Dimitri Van De Ville, *rapporteur*
Prof. Nick Kingsbury, *rapporteur*
Prof. Jean-Christophe Pesquet, *rapporteur*

École polytechnique fédérale de Lausanne—2011

Abstract

It is well-known that the Gaussian functions and, more generally, their modulations-translations (the Gabor functions) have the unique property of being optimally localized in space and frequency in the sense of Heisenberg's uncertainty principle. In this thesis, we address the construction of complex wavelets modeled on the Gabor function, and smoothing kernels based on the Gaussian. We proceed by relaxing the exact form of the Gaussian and Gabor function, and by approximating them using spline functions. In particular, we construct a family of spline wavelets, termed Gabor-like wavelets, which provide arbitrary close approximations of the Gabor function. On the other hand, we introduce a family of compactly supported box splines to approximate the Gaussian, both isotropic and anisotropic. The attractive feature of these spline wavelets and kernels is that we are able to develop fast and efficient algorithms for implementing the associated transforms.

The Gabor-like wavelet is obtained within the framework of multiresolution analysis by combining Hilbert transform pairs of B-spline wavelets. To begin with, we provide a rigorous understanding of why the Hilbert transform goes well with wavelets. We show that at the heart of this is the characteristic vanishing-moment property of wavelets and certain fundamental invariances of the Hilbert transform. The former allows us to ensure that the Hilbert transform (which is non-local) of a localized wavelet is again well-localized provided that it has sufficient number of vanishing moments, while the latter allows us to seamlessly integrate it into the multiresolution framework of wavelets. Guided by these facts, we formulate a general recipe for constructing a pair of wavelet bases that form a Hilbert transform pair. Using this recipe, we are able to identify a pair of B-spline wavelets that are related through the Hilbert transform. We show that the complex wavelet derived from this pair converges to a Gabor function as its order gets large. We next extend the construction to higher dimensions using the directional Hilbert transform and tensor-products wavelets. This results in a system of complex wavelets that closely resemble the directional Gabor functions. We develop an efficient numerical algorithm for implementing the associated complex wavelet transforms on finite periodic data.

We next identify the complete family of transforms which share the fundamental invariances of the Hilbert transform. Based on this family of transforms and its particular properties, we are able to provide an amplitude-phase interpretation of the signal representation associated with the Gabor-like wavelet transform. This allows us to understand the significance of the amplitude and phase information associated with the transform.

As an application, we develop a coarse-to-fine stereo-matching algorithm that does dynamic programming on the sub-sampled Gabor-like wavelet pyramid instead of the raw pixel intensities. The crucial feature of our pyramid was that it provides near translation-invariance at the cost of moderate redundancy. The translation-invariance is absolutely essential for encoding the local spatial translations between the stereo pair. Based on the particular Gabor-like form of our wavelets, we also provided a mathematical explanation

of the near translation-invariance of our pyramid. From a computational standpoint, we show that a significant reduction of the run time is achieved in comparison with the standard dynamic programming algorithm.

In the second half of the thesis, we introduce a particular bivariate box spline termed the radially-uniform box spline. As an application of the Central Limit Theorem, we show that it converges to a Gaussian as its order gets large. For a fixed order, we show how the parameters of the box spline can be tuned to approximate a fixed anisotropic Gaussian. In particular, we develop a simple root-finding algorithm for controlling the anisotropy of the elliptical box splines.

We next investigate the efficient realization of space-variant (or non-convolution) Gaussian filters using these box splines. The realization of even the simple convolution Gaussian filter is known to be computationally challenging, particularly when the size of Gaussian is large. The number of computations required per pixel for a direct implementation of the filter scales linearly with the size of the filter. We demonstrated that it is possible to filter an image with Gaussian-like box splines of varying size, elongation and orientation using a fixed number of computations per pixel (constant-time implementation). The associated algorithm is easy to implement and uses simple pre-integrations and local finite-differences.

As an application of the Gaussian-like box splines and the associated filtering algorithm, we develop two algorithms for space-variant filtering. The first of these is inspired by anisotropic Gaussian diffusion. The space-variance in this case is in terms of the size, elongation, and orientation of the box splines, which are controlled using the local image features. The other scheme is based on a space-variant form of the Gaussian bilateral filter. The spatial adaptability in this case was in terms of the size of the spatial Gaussian filter. The highlight is that we are able to develop a constant-time algorithm for implementing the bilateral filter by approximating the variable spatial filter using isotropic box splines, and by approximating the fixed range filter (locally) using a class of shiftable kernels. We demonstrate their usage by developing smoothing algorithms for signal-adaptive denoising of images. As an application in a different direction, we develop box spline filters resembling the Laplacian-of-Gaussian. Using this particular detector, and by appropriately modifying the basic filtering algorithm, we develop a fast template-matching algorithm for the detection of bright cells and nuclei in fluorescence images.

Keywords

Résumé

Les fonctions Gaussiennes, et plus généralement les versions modulées et décalées de celles-ci (i.e., fonctions de Gabor), sont connues notamment en vertu de leur propriété unique d'être optimalement localisées dans les domaines spatial et fréquentiel, au sens du principe d'incertitude d'Heisenberg. Dans le cadre de cette thèse, nous considérons ainsi la construction d'ondelettes complexes et de noyaux de lissage basés sur les fonctions de Gabor et sur les Gaussiennes, respectivement. Pour ce faire, nous relaxons la forme exacte de ces fonctions et les approximons par des splines. Spécifiquement, nous construisons d'une part une famille d'ondelettes de type spline, appelées ondelettes 'à la Gabor', qui fournissent une approximation contrôlable des fonctions éponymes. D'autre part, nous introduisons des familles de splines 'box' à support compact, aussi bien isotropes qu'anisotropes, permettant d'approximer la Gaussienne. L'intérêt de ces ondelettes et noyaux de type spline réside entre autres dans la possibilité de développement d'algorithmes rapides implémentant les transformées associées.

Dans le cadre de l'analyse multi-résolution, l'ondelette 'à la Gabor' est obtenue en combinant des paires de Hilbert d'ondelettes B-splines. Tout d'abord, nous proposons une explication rigoureuse quant à la compatibilité de la transformée de Hilbert avec les transformées en ondelettes. Nous montrons qu'au coeur de cette compatibilité résident les moments nuls spécifiques aux ondelettes, ainsi que certaines invariances fondamentales de la transformée de Hilbert. Etant donné une fonction d'ondelette localisée comportant un nombre suffisant de ces moments nuls, il est ainsi possible de garantir que sa transformée de Hilbert (dont l'effet associé est généralement non-local) sera également localisée. De plus, les invariances susmentionnées impliquent de par leur nature l'intégration directe de ladite fonction dans la représentation multi-résolution propre aux ondelettes. Forts de ces résultats, nous développons une méthodologie globale permettant de construire une paire de bases d'ondelettes formant une paire de Hilbert. Ce cadre nous permet de définir une paire d'ondelettes B-spline liées par la transformée de Hilbert. Nous montrons qu'une ondelette complexe ainsi obtenue converge vers une fonction de Gabor lorsque son ordre croît. Puis, nous étendons cette construction unidimensionnelle aux dimensions supérieures en utilisant la transformée de Hilbert directionnelle et les produits tensoriels d'ondelettes. Les ondelettes complexes résultantes forment un système d'analyse proche des fonctions de Gabor directionnelles. Nous développons un algorithme numérique efficace implémentant les transformées associées sur des données finies et périodiques.

Nous identifions ensuite la famille complète de transformées partageant les invariances fondamentales de la transformée de Hilbert. Basés sur cette famille et ses propriétés propres, nous sommes à même de fournir une interprétation de la représentation du signal en termes de phase et d'amplitude qui soit associée à la transformée en ondelettes 'à la Gabor'. Cela nous permet ainsi de donner une signification aux valeurs d'amplitude et de phase résultant de la transformée en ondelettes complexes.

En tant qu'application de ces méthodes, nous développons un algorithme hiérarchique ('coarse-to-fine') de correspondance stéréo. Son fonctionnement repose sur une programmation dynamique utilisant une pyramide 'à la Gabor' sous-échantillonnée plutôt que l'intensité brute des pixels. La caractéristique clef de notre pyramide est sa quasi-invariance par translation en dépit de sa faible redondance ; l'invariance par translation étant indispensable pour encoder les translations spatiales locales entre les paires d'images stéréo. Nous fournissons de plus une explication mathématique de la quasi-invariance par translation de notre pyramide en se basant sur la forme 'à la Gabor' de nos ondelettes. D'un point de vue computationnel, nous montrons que le temps de calcul pour notre méthode est substantiellement réduit par rapport à l'algorithme standard de programmation dynamique.

Dans la seconde moitié de cette thèse, nous introduisons une forme bivariée de la spline 'box' que nous appelons spline 'box' radialement uniforme. Appliquant le théorème central limite, nous prouvons que cette dernière converge vers une Gaussienne lorsque son ordre croît. Nous montrons en outre comment les paramètres d'une spline 'box' d'ordre donné peuvent être ajustés afin d'approximer une fonction Gaussienne anisotrope donnée. En particulier, nous développons un algorithme simple de recherche de racines ayant pour but de contrôler l'anisotropie des splines 'box' elliptiques.

A partir de ces splines 'box', nous nous intéressons ensuite à la construction de filtres Gaussiens variables spacialement (i.e., non-convolutifs). L'application d'un filtre Gaussien - fût-il convolutif - est notoirement coûteuse en temps de calcul, surtout lorsque la taille du filtre en question est importante. Dans une implémentation directe de ce filtre, le nombre correspondant de calculs par pixel est en effet proportionnel à cette taille. Nous démontrons cependant qu'une opération de filtrage utilisant des splines 'box' de type Gaussien ne nécessite qu'un nombre fixe d'opérations par pixel (implémentation à temps constant), et ce pour une taille, une élongation et une orientation variables. L'algorithme associé est aisé à implémenter ; il utilise de simples pré-intégrations et différences finies locales.

Ces splines 'box' de type Gaussien ainsi que l'algorithme de filtrage associé nous ont conduit au développement de deux algorithmes effectuant un filtrage spatialement variable. Le premier de ces algorithmes s'inspire du phénomène de diffusion Gaussienne anisotrope. Dans ce contexte, la variabilité spatiale s'exprime en termes de taille, d'élongation et d'orientation des splines 'box', ces paramètres s'adaptant aux caractéristiques locales des images. Le second algorithme est basé sur une extension spatialement variable du filtre Gaussien bilatéral. Dans ce cas-ci, la variabilité spatiale est paramétrée par la taille du filtre Gaussien. Soulignons que notre algorithme de filtrage bilatéral a pour point fort d'être à temps constant. Dans ce but, le filtre spatialement variable est approximé par des splines 'box', tandis que le filtre fixe de gamme d'intensité est (localement) approximé à l'aide d'une classe de noyaux spatialement invariants. Nous illustrons l'utilisation de ces méthodes en développant des algorithmes de lissage appliqués au débruitage adaptatif d'images. En tant qu'application distincte, nous construisons des filtres splines 'box' approchant le Laplacien d'une Gaussienne. En utilisant ce détecteur particulier, et en modifiant en conséquence l'algorithme de filtrage initial, nous développons un algorithme rapide de correspondance de modèle que nous utilisons pour la détection de cellules et de noyaux dans des images de fluorescence.

Mots-clé

fonction Gaussienne, fonction de Gabor, localisation optimale, ondelettes complexes, transformée de Hilbert, invariances, analyse multi-résolution, 'shiftability', représentation en amplitude-phase, correspondance stéréo, filtrage spacialement variable, splines 'box', diffusion Gaussienne, filtre bilatéral, détection de cellules, algorithme en temps constant, la complexité $O(1)$.

Acknowledgement

To my family.

General notation

We collect the general notations and terminology used in the thesis.

The set of integers and real numbers are denoted by $\mathbf{Z}$ and $\mathbf{R}$, respectively. We use bold letters to distinguish vectors from scalars. For example, $f(x)$ denotes a function of the real-valued variable x, while $f(\boldsymbol{x})$ is a function of the multivariate $\boldsymbol{x} = (x_1, \ldots, x_d)$ which lives in $\mathbf{R}^n$. We use round brackets $(\cdot)$ for continuous variables, and square brackets $[\cdot]$ for discrete variables. For example, $f[n]$ will denote a sequence defined on the integers, while $f[\boldsymbol{n}]$ will typically be defined on $\mathbf{Z}^n$. We use bold capital letters to represent matrices, while their entries will either be denoted by subscripted lower case letters, e.g., $\mathbf{A} = (a_{i,j})$, or with brackets, e.g., $\mathbf{B} = (\mathbf{B}(i,j))$. We denote the identity matrix by $\mathbf{I}$.

We use j to denote the unit $\sqrt{-1}$. The conjugate of a complex number z is denoted by z^*. The real and imaginary components of z are denoted by $\mathbf{Re}(z)$ and $\mathbf{Im}(z)$, while the modulus and argument (phase) are denoted by $|z|$ and $\arg(z)$ (or $\angle z$). We use z^γ to denote the γ-th power of some complex number z, which is defined to be $|z|^\gamma e^{j\gamma \arg(z)}$ where $-\pi < \arg(z) < \pi$ is the principal argument of z. On this principal branch, the basic identity $(z_1 z_2)^\gamma = z_1^\gamma z_2^\gamma$ holds only if $\arg(z_1) + \arg(z_2) \in (-\pi, \pi)$.

We use $\hat{f}(\boldsymbol{\omega})$ to denote the *Fourier transform* of a function $f(\boldsymbol{x})$ on $\mathbf{R}^n$. This is defined by

$$\hat{f}(\boldsymbol{\omega}) = \int_{\mathbf{R}^n} f(\boldsymbol{x}) \exp(-j\boldsymbol{\omega}^T \boldsymbol{x}) \, d\boldsymbol{x} \qquad (\boldsymbol{\omega} \in \mathbf{R}^n)$$

where $\boldsymbol{\omega}^T \boldsymbol{x} = \sum_{i=1}^n \omega_i x_i$ is the inner-product between the frequency variable $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_d)$ and the space variable $\boldsymbol{x} = (x_1, \ldots, x_d)$. We suppress the domain of an integral (or summation) if this is obvious from the context.

We use $f(\cdot - \boldsymbol{s})$ to denote the function obtained by translating $f(\boldsymbol{x})$ by $\boldsymbol{s}$. The *convolution* of functions $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ is given by

$$(f * g)(\boldsymbol{x}) = \int f(\boldsymbol{s})g(\boldsymbol{x} - \boldsymbol{s})d\boldsymbol{s}.$$

We often simply write this as $f * g(x)$. The notation $\circledast_{k=1}^N f_k(\boldsymbol{x})$ is used to denote the convolution of a collection of functions $f_1(\boldsymbol{x}), \ldots, f_N(\boldsymbol{x})$. The order of the convolutions will be immaterial for the functions of interest.

We denote the first derivative of $f(x)$ by one of the following symbols: $f'(x), Df(x)$, or $df(x)/dx$. In general, the k-th derivative is denoted by $f^{(k)}(x)$. We say that $f(x)$ is n-times continuously differentiable if all its derivatives up to order n exists and are continuous. For a multivariate function $f(x) = f(x_1, \ldots, x_n)$, the partial derivative along x_i is often represented by $\partial_i f(x)$. A function $f(x)$ is said to be *Lipschitz continuous* (or simply Lipschitz) if there exists a constant K such that

$$|f(x) - f(y)| \leq K|x - y| \qquad (x, y \in \mathbf{R}).$$

The smallest such constant is the Lipschitz norm of $f(x)$ and is denoted by $\|f\|_{\text{Lip}}$. We write $f(x) = O(g(x)), x \in A$, to signify that $|f(x)| \leq Cg(x)$ for all $x \in A$, where C is an absolute constant.

We use $\mathbf{M}^n$ to denote the $(n-1)$-fold matrix multiplication of $\mathbf{M}$ with itself. The integral $\int \mathbf{M}(x) f(x) \, dx$, corresponding to a real-valued function $f(x)$ and a matrix-valued function $\mathbf{M}(x)$, represents a matrix of the same dimension as $\mathbf{M}(x)$, whose (i, j)-th component is given by $\int \mathbf{M}_{i,j}(x) f(x) \, dx$. If $\mathbf{P}$ and $\mathbf{Q}$ are constant matrices, we then have $\int \mathbf{P}\mathbf{M}(x)\mathbf{Q} f(x) \, dx = \mathbf{P}(\int \mathbf{M}(x) f(x) \, dx)\mathbf{Q}$.

We use $T_1 \circ T_2$ (often simply $T_1 T_2$) to denote the composition of two transformations (or operators) T_1 and T_2, that is, $(T_1 \circ T_2)(f) = T_1(T_2(f))$ for every element f in that domain. The order of the composition is important; in general, $T_1 T_2 \neq T_2 T_1$. We use $\mathscr{I}$ to denote the identity operator on function spaces.

We define the function $\text{sinc}(x)$ to be $\sin(x)/x$ at all non-zero x, and as zero at the origin (non-conventional definition). The Dirac delta distribution is denoted by $\delta(x)$. The Kronecker delta function is denoted by $\delta[n]$. The *signum* function, denoted by $\text{sign}(x)$, equals 1 when $x > 0$, equals -1 when $x < 0$, and zero when $x = 0$.

We use standard notations for norms, that is,

$$\|f\|_p = \left[\int |f(x)|^p \, dx \right]^{1/p} \qquad (1 \leq p < \infty)$$

and

$$\|f\|_\infty = \text{ess sup} \left\{ |f(x)| : x \in \mathbf{R} \right\}.$$

The space of functions having finite p-norm is denoted by $\mathbf{L}^p(\mathbf{R}^n)$, or simply, by $\mathbf{L}^p$. We explicitly use $\ell^p(\mathbf{Z})$, or ℓ^p, to denote the $\mathbf{L}^p$ spaces corresponding to sequences. An operator T which takes an $\mathbf{L}^p$ function into an $\mathbf{L}^q$ is said to be *bounded* if, for some absolute constant C,

$$\|Tf\|_q \leq C\|f\|_p \qquad (f \in \mathbf{L}^p).$$

We almost always work in $\mathbf{L}^2(\mathbf{R}^n)$, the Hilbert space of *finite-energy* (or *square-integrable*) functions. This is equipped with the inner product

$$\langle f, g \rangle = \int f(x) g(x) \, dx.$$

Two functions $f(x)$ and $g(x)$ are said to be *orthogonal* if $\langle f, g \rangle = 0$. The notation $\mathbf{W}^{2,\gamma}(\mathbf{R})$ is used to denoted the fractional *Sobolev space* of order γ, namely, the functions in $\mathbf{L}^2(\mathbf{R})$ for which

$$\int (1 + |\omega|^2)^\gamma |\hat{f}(\omega)|^2 \, d\omega < \infty.$$

The corresponding norm is given by the square-root of the above quantity, and is denoted by $\|f\|_{\mathbf{W}^{2,\gamma}}$. For integer γ, this is simply the collection of functions having (weak) derivatives up to order γ, such that the function along with all its derivatives are in $\mathbf{L}^2$.

The class of infinitely differentiable functions that are rapidly decreasing at infinity along with all partial derivatives (Schwartz class) is denoted by $\mathbf{S} = \mathbf{S}(\mathbf{R})$. The space of continuous functionals on $\mathbf{S}$ is denoted by $\mathbf{S}'$.

We denote a countable sequence of functions by $(f_n)_{n \in \mathbf{Z}}$ or simply by (f_n). The ℓ^2-*span*, or simply the span, of a sequence (f_n) is the set of all linear combinations of (f_n) weighted with ℓ^2 sequences, that is,

$$\text{span}(f_n) = \Big\{ \sum_{n \in \mathbf{Z}} c_n f_n, \ \text{where} \ \sum_{n \in \mathbf{Z}} |c_n|^2 < \infty \Big\}.$$

A sequence (f_n) is said to be a *basis* for $\mathbf{L}^2$ if $\mathbf{L}^2 = \text{span}(f_n)$, and if $\sum_i c_i f_i = 0$ implies $\boldsymbol{c} = 0$ for every finite sequence $\boldsymbol{c} = (c_n)$.

A sequence (f_n) is said to be a *Riesz basis* for $\mathbf{L}^2$ if it is topologically isomorphic to some (and hence every) orthonormal basis of $\mathbf{L}^2$. Equivalently, this means that it is complete in $\mathbf{L}^2$ (i.e., $\text{span}(f_n) = \mathbf{L}^2$), and that there exists positive constants A and B such that

$$A \sum_{n=1}^{N} |c_n|^2 \leq \Big\| \sum_{n=1}^{N} c_n f_n \Big\|_2^2 \leq B \sum_{n=1}^{N} |c_n|^2$$

for every integer N and scalars $c_1, \ldots, c_N$.

We use $f_{i,k}(x)$ to denote the combined operations of (dyadic) dilation and translation used in wavelet theory,

$$f_{i,k}(x) = 2^{i/2} f(2^i x - k) \qquad (i \in \mathbf{Z}, k \in \mathbf{Z}).$$

Contents

Chapter 1

Introduction

1.1 Natural atoms for signal processing

THE Uncertainty Principle in signal theory loosely refers to the principle that *"the more we locate a signal in the space domain, the less we can locate it in the frequency domain, and vice versa"*. While an exact mathematical interpretation of this principle can be formulated in different ways, the most significant is the one due to Heisenberg [1]. In its simplest form, the principle asserts that for any unit-energy waveform $f(x)$,

$$\left(\int |x f(x)|^2 \, dx \right) \left(\int |\omega \hat{f}(\omega)|^2 \, d\omega \right) \geq \frac{1}{4}.$$

The first and the second terms on the left are respectively the measure of the localization (concentration) of the waveform in space and frequency. Their product quantifies the localization of the waveform in the joint space-frequency domain. It is well-known that the functions which attain the lower bound is the Gaussian, and more generally, its space-frequency translates, the so-called Gabor functions [2]:

$$\frac{1}{\sigma \sqrt{2\pi}} \exp \left(-\frac{(x - x_0)^2}{2\sigma^2} \right) \exp \left(j \omega_0 (x - x_0) \right).$$

Apart from this optimality criterion, there is also considerable evidence that images in primary visual cortex are represented in terms of hierarchically arranged Gabor functions [3]. Due to these reasons, the Gabor function is used as the natural atom for signal processing. The lowpass Gaussian, on the other hand, is ubiquitously used as a smoothing kernel in image processing.

1.1.1   Spatial adaptability

A shortcoming of standard Gabor analysis and Gaussian smoothing is that they use a window of fixed size for processing signals. The former uses the translates of a fixed Gaussian window in space and frequency, while smoothing amounts to translating the window in space.

This thesis aims at providing better spatial adaptability in either case. We improve the spatial adaptability of Gabor analysis using the framework of multiresolutions and wavelets. This is achieved through the realization of a family of complex wavelets that provide arbitrary close approximations of the Gabor function. Using the associated wavelet transform, we are able to analyze signals using the dilations and translations of these Gabor-like wavelets, and then reconstitute them back from the analysis coefficients using a fast and stable algorithm. The final point is particularly important from a computational standpoint, since the reconstruction process in Gabor representations is known to be computationally expensive and often unstable [4, 5]. The key aspect of our scheme is that, instead of the uniform space-frequency tiling used in Gabor analysis, we use the non-uniform space-frequency tiling used in wavelet theory. As is well-known, the latter "variable-window" analysis has the advantage that it allows us to locate high-frequency components more accurately in space.

On the other hand, to improve the adaptability of Gaussian smoothing, we introduce a family of bivariate splines called box splines [6]. We show that they can be use to approximate both isotropic and anisotropic Gaussians, and that we can control their shape and size using a small number of parameters. In particular, we demonstrate that it is possible to filter an image using these Gaussian-like box splines of variable shape and size at a remarkably low complexity.

1.1.2   Fast algorithms using B-splines

It is well-known that B-splines yield fast algorithms for signal processing [7, 8]. The common point of our construction is that we relax the exact Gabor or Gaussian form of the kernel, and instead approximate them using B-splines. This is based on the idea that one can arbitrarily approximate the Gaussian using higher order B-splines, which are obtained by the repeated convolution of the box function with itself [7]. This can well be seen as a manifestation of the Central Limit Theorem in that one can approximate a Gaussian by convolving a sufficiently large number of regular "bump" functions. The Gabor-like wavelets, in turn, are obtained by appropriately modulating the Gaussian-like B-splines. The advantage with this relaxation is that we are able to derive fast and efficient algorithms for implementing the variable-window Gabor analysis and Gaussian smoothing.

Before explaining the main ideas in more detail, we demystify a characteristic "scaling" property of B-splines that lies at the heart of the thesis. The simplest

B-spline is the box (or rect) function, given by

$$\beta^0(x) = \begin{cases} 1 & \text{if } 0 \le x < 1 \\ 0 & \text{otherwise .} \end{cases} \tag{1.1}$$

This is the B-spline of degree 0. Higher-degree B-splines, having better smoothness, are obtained from the box function through convolutions [9]. In particular, the B-spline of degree m is specified by the induction

$$\beta^m(x) = (\beta^{m-1} * \beta^0)(x) \qquad (m \ge 1). \tag{1.2}$$

The scaling properties that we are alluding to can be read-off easily from the Fourier transform of (1.1):

$$\hat{\beta}^0(\omega) = \frac{1 - \exp(-j\omega)}{j\omega}. \tag{1.3}$$

This shows us that

$$\hat{\beta}^0(2\omega) = \frac{1}{2}\left(1 + \exp(-j\omega)\right)\frac{1 - \exp(-j\omega)}{j\omega} = \frac{1}{2}\left(1 + \exp(-j\omega)\right)\hat{\beta}^0(\omega). \tag{1.4}$$

Expressed in space domain, this translates in to the so-called "two-scale" relation:

$$\frac{1}{2}\beta^0\left(\frac{x}{2}\right) = \frac{1}{2}\left[\beta^0(x) + \beta^0(x-1)\right].$$

This tells us that we can dilate the box function simply by taking the linear combination of its translates. The more subtle fact is that this generalizes directly to the higher-degree B-splines in (1.2). This is seen by noting that

$$\widehat{\beta^m}(\omega) = \left(\frac{1 - \exp(-j\omega)}{j\omega}\right)^{m+1},$$

and that

$$\widehat{\beta^m}(2\omega) = \frac{1}{2^{m+1}}\left(1 + \exp(-j\omega)\right)^{m+1}\widehat{\beta^m}(\omega).$$

As a result, we have the two-scale relation

$$\frac{1}{2}\beta^m\left(\frac{x}{2}\right) = \sum_{n=0}^{m+1} h[n]\beta^m(x-m) \tag{1.5}$$

where

$$h[n] = \begin{cases} \dfrac{1}{2^{m+1}}\dbinom{m+1}{n} & \text{for } n = 0, 1, \ldots, m+1 \\ 0 & \text{otherwise.} \end{cases}$$

The filter $h[n]$ is referred to as the binomial filter.

The two-scale relations can be used to derive fast algorithms for rescaling signals expressed in a B-spline basis [8]. In particular, it is these two-scale relations that allow us to conceive the B-spline multiresolution, and subsequently, the B-spline wavelets. The B-splines, in fact, are part of a larger class of functions that satisfy relations similar to (1.5). They form the fundamental constructs in wavelet theory, and are termed as "scaling functions". We discuss this connection in detail in §1.2.

There is also a complementary form of the scaling relation in (1.4), which we use for deriving fast algorithms for filtering images using Gaussians of variable shape and size. This is precisely the relation

$$\hat{\beta}^0(a\omega) = \frac{1 - \exp(-ja\omega)}{a} \left(1 - \exp(-j\omega)\right)^{-1} \hat{\beta}^0(\omega), \qquad (1.6)$$

where $a > 0$ is any arbitrary real number. In the space domain, this can be expressed as

$$\frac{1}{a}\beta^0\left(\frac{x}{a}\right) = \Delta_a \Delta^{-1} \beta^0(x) \qquad (1.7)$$

where

$$\Delta_a f(x) = \frac{1}{a}\left(f(x) - f(x-a)\right) \qquad \text{and} \qquad \Delta^{-1} f(x) = \sum_{k=0}^{\infty} f(x-k).$$

We call $\Delta_a f(x)$ the finite-difference of $f(x)$, and $\Delta^{-1} f(x)$ the running-sum or pre-integration.

The significance of (1.7) is that we can rescale the box function by any arbitrary factor simply by taking its (non-local) running-sum followed by its (local) finite-difference. The crucial point is that it is only the local operation that depends on the scale. As a result, by pre-integrating a signal in a B-spline basis, we are able to rescale it by different factors simply by applying appropriate finite-differences. As explained in detail in §1.3, this can be used to derive fast and efficient algorithms for space-variant filtering in which the scale (shape or size) of the filter is changed from point-to-point.

1.2   Multiresolution approximation and wavelets

Our construction of Gabor-like complex wavelets, and the associated algorithm for analyzing and reconstituting the signal, is based on the theory of multiresolution analysis of Mallat and Meyer [10, 11]. The simplest forms of multiresolution analyses are generated using splines. They also provide fast algorithms for computing the multiresolution approximations [12, 13, 14]. To explain this, we considering the "discrete" box function, and the associated piecewise-constant multiresolution. This is the system of nested approximation spaces

$$\mathbf{V}_0 \supset \cdots \supset \mathbf{V}_1 \supset \cdots,$$

where the signals in $\mathbf{V}_i$ are constant over blocks of the form $2^i n, \ldots, 2^i(n+1)-1$, where $n \in \mathbf{Z}$. The information in this case is precisely the height of the signal in each block, and we denote it by $f_i[n]$. The signals in $\mathbf{V}_i$ are said to be of resolution 2^i.

Given a signal $f_0[n]$ in $\mathbf{V}_0$, we want to approximate it by a signal having resolution 2^i. The signal at this resolution which is "closest" to $f_0[n]$ in the least-square sense is given by the orthogonal projection of $f_0[n]$ onto $\mathbf{V}_i$. It is straightforward to verify that the projected signal $f_i[n]$ is given by

$$f_i[n] = \frac{1}{2^i} \sum_{k=0}^{2^i-1} f_0[2^i n + k] \qquad (n \in \mathbf{Z}). \tag{1.8}$$

Note that, a direct computation of the multiresolution approximation using (1.8) requires a total of 2^i basic operations per sample. In other words, the computational complexity scales *exponentially* with the level of the approximation. The remarkable fact is that we can bring down the complexity from $O(2^i)$ to $O(1)$ using the *pyramid* algorithm[1] [15, 16, 17]. The connection of this algorithm with (1.8) is seen by observing that the approximations at successive resolutions are related through

$$f_{i+1}[n] = \frac{1}{2}(f_i[2n] + f_i[2n+1]). \tag{1.9}$$

Thus, starting from the signal approximation at some arbitrary resolution, we can obtain the approximation at the next resolution using exactly two operations per sample (cf. Figure 1.1). In other words, we can recursively compute the multiresolution approximations $f_1[n], f_2[n], \ldots$ using a fixed number of operations per sample.

1.2.1 B-spline multiresolution

The multiresolution we just discussed involved discrete signals. It can be extended to the class of continuously-defined finite-energy signals. This results in the famous Haar multiresolution [18]. In this case, $\mathbf{V}_i$ is the space of functions which are constant over the interval $[n2^i, (n+1)2^i], n \in \mathbf{Z}$. We can write every $f_0(x)$ belonging to $\mathbf{V}_0$ as

$$f_0(x) = \sum_{n \in \mathbf{Z}} f_0[n] \beta^0(x - n)$$

It can be verified that the orthogonal projection of $f_0(x)$ onto $\mathbf{V}_i$ is given by

$$f_i(x) = \sum_{n \in \mathbf{Z}} f_i[n] \beta^0(2^{-i} x - n)$$

---

[1]The pyramid algorithm was introduced between 1981-1983 by Burt, Adelson, and Crowley in computer vision to represent real-world signals and images at different resolutions or scales. The idea was to adapt the signal resolution to process only the relevant details for a given task.

where the coefficients $f_i[k]$ are derived from $f_0[k]$ using (1.8).

In the present case, the signal approximations are given by discontinuous functions. We can extend this framework to include smooth approximations as follows. To do so, we fix some $m \geq 1$, and let $\mathbf{V}_i$ be the space of $(m-1)$ times continuously differentiable and square-integrable functions, whose restriction on any interval of the form $[n2^i, (n+1)2^i]$ is a polynomial of degree m. Clearly, $\mathbf{V}_{i+1} \subset \mathbf{V}_i$. A fundamental result in spline theory is that a (Riesz) basis of $\mathbf{V}_0$ is given by the translates of the *B-spline* $\beta^m(x)$ of degree m [9]. This gives us the B-spline multiresolution

$$\{0\} \subset \cdots \subset \mathbf{V}_1 \subset \mathbf{V}_0 \subset \mathbf{V}_{-1} \subset \cdots \subset \mathbf{L}^2(\mathbf{R})$$

where

$$\mathbf{V}_i = \text{span}\left\{\beta^m(2^{-i}x - k): k \in \mathbf{Z}\right\}.$$

The nested structure of the approximation spaces is confirmed by the two-scale relation in (1.5). Starting from the approximation

$$f_0(x) = \sum_{n \in \mathbf{Z}} f_0[n]\beta^m(x-n)$$

the multiresolution approximations (orthogonal projections) in $\mathbf{V}_1, \mathbf{V}_2, \cdots$ can be obtained through the fast pyramid algorithm described earlier. For $m = 0$, we exactly recover (1.9). When $m > 0$, the simple averaging in (1.9) must be replaced by a more general filtering [12].

1.2.2   Gabor-like wavelets

In going from the original to the coarser resolution, we incur a loss of information. This is given by the projection error $f_0(x) - f_1(x)$. Instead of keeping the signals at both resolutions, it is possible to encode the error (or residual) using a *difference pyramid* [15, 17, 19]. The problem with such representations is that they have more degrees of freedom than is necessary. An optimal representation is, in fact, provided by the wavelet construction of Mallat and Meyer [10].

This brings us to the so-called *B-spline wavelet* $\psi^m(x)$ [12]. A fundamental result is that the projection error can be encoded in terms of the translates of the wavelet $\psi^m(2^{-1}x)$:

$$f_0(x) - f_1(x) = \sum_{n \in \mathbf{Z}} d_1[n]\psi^m(2^{-1}x - n).$$

In other words, we have the decomposition

$$f_0(x) = \sum_{k \in \mathbf{Z}} f_1[k]\beta^m(2^{-1}x - k) + \sum_{n \in \mathbf{Z}} d_1[n]\psi^m(2^{-1}x - n).$$

We can now iterate this up to a given resolution 2^J. This results in the wavelet representation

$$f_0(x) = \sum_{k \in \mathbf{Z}} f_J[k]\beta^m(2^{-J}x - k) + \sum_{i=1}^{J} \sum_{n \in \mathbf{Z}} d_i[n]\psi^m(2^{-i}x - n).$$

The wavelet coefficients $d_i[n], 1 \leq i \leq J$, are obtained from the coefficients $f_{i-1}[k]$ of the finer multiresolution using Mallat's filterbank algorithm [12].

The attractive property of the B-spline wavelet $\psi^m(x)$ is that it converges to the real part of the Gabor function as m gets large. This was established by Unser et al. in [20]. In this thesis, we show how the imaginary part of the Gabor function can be approximated using a larger class of wavelets called fractional B-spline wavelets [21, 22]. The imaginary part is obtained by applying the *Hilbert transform* to the B-spline wavelet. This does not come as a surprise since it is well-known that the real and imaginary components of the Gabor function form (approximate) Hilbert transform pairs. By combining the B-spline wavelet and its Hilbert transform, we get complex wavelet, termed *Gabor-like wavelet*, that has near-optimal space-frequency localization.

There is also a complementary aspect about the Gabor-like wavelet. It is a well-known fact that real wavelet transforms lack translation-invariance—simple translation of the input signal does not produce a simple translation of the wavelet coefficients [23, 14]. It was shown by Kingsbury and Selesnick [24, 25, 26] that the poor translation-invariance of standard wavelet bases can be improved by considering a quadrature pair of wavelet bases, where the mother wavelets are related through the Hilbert transform. The idea was to realize a complex wavelet transform by combing the pair of "matched" wavelets. It was empirically found that modulus of the coefficients of the resulting complex wavelet transform had better translation-invariance than the standard wavelet transform [27]. This also holds for the complex wavelet transform associated with the Gabor-like wavelets. We provide a mathematical explanation of the improvement of the translation-invariance using the fact that the wavelets in our case closely resemble the Gabor function. This is used to develop an efficient coarse-to-fine algorithm for stereo matching .

1.3 Space-variant filtering

The crucial property of B-splines which leads to multiresolutions, and eventually to wavelets, is the two-scale relation (1.5). Using similar "scaling" properties, one can also derive fast algorithms for fast Gaussian filtering [28, 29, 30]. This exploits the fact that the averaging process uses translates of the filter which have significant overlaps. To convey the idea, we consider the averaging of a signal $f(x)$ using the symmetric box function

$$\beta_a(x) = \begin{cases} 1/a & \text{if } -a/2 \leq x < a/2 \\ 0 & \text{otherwise .} \end{cases}$$

This has a total mass of unity. The averaged signal is given by

$$\overline{f}(x) = \frac{1}{a} \int_{-a/2}^{a/2} f(y)\, dy. \tag{1.10}$$
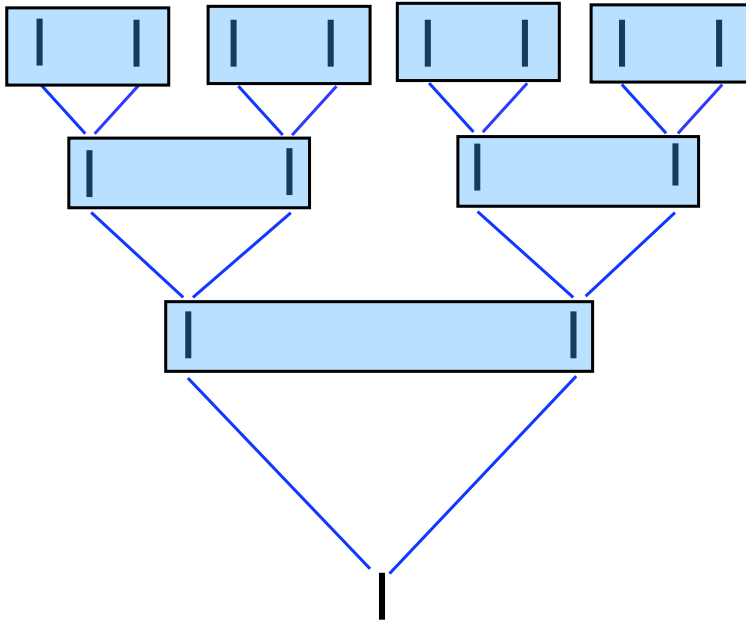
Pyramid decomposition

Figure 1.1: Schematic of classical pyramid algorithm for the simplest multiresolution approximation, the "piecewise-constant" or Haar multiresolution. A 3-level decomposition of an 8-point signal is computed in this case. The black bars denote the signal samples. The topmost level corresponds to the signal at the original resolution ($i = 0$). At every successive level, the sample at each point is derived by averaging two samples (within the blue window) from the finer resolution. Note that the signal at a coarse resolution $i \geq 1$ is, in fact, given by the average of 2^i samples of the original signal. This direct computation of the coarse resolution, however, requires 2^i basic operations per sample. As against this, the pyramid algorithm recursively computes the samples at every coarse resolution using just two operations per sample.
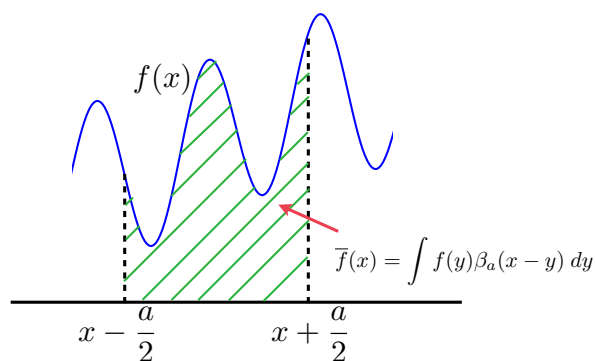
Figure 1.2: Computation of moving-average by convolving it with the box function. To compute $\bar{f}(x)$, we first localize the signal using the box function translated to x. We then compute the area of overlap (hatched zone). The total number of computations is clearly proportional to the size of support a. The non-trivial fact is that this can be done in constant-time by first computing the primitive of $f(x)$ and then taking its finite-difference as per (1.11) and (1.12). This idea is also reflected in (1.7).

Note that the number of operations required per pixel for a direct computation of (1.10) is proportional to a. The non-trivial fact is that we can compute (1.10) using a fixed number of computations per position, independent of a. This is clear once we write (1.10) as

$$\bar{f}(x) = \frac{1}{a}\left[F(x + a/2) - F(x - a/2)\right] \tag{1.11}$$

where

$$F(x) = \int_{-\infty}^{x} f(y)\, dy. \tag{1.12}$$

Clearly, the computation of $F(x)$ is independent of a, while (1.11) requires one addition and multiplication per position (cf. Figure 1.2). This is the basic idea behind constant-time averaging. We note that this can also be read-off from (1.7)— the running-sum in (1.7) is precisely the indefinite integral in (1.12) applied to piecewise-constant functions in the B-spline basis.

As in the case of the pyramid, the above idea can directly be generalized to the higher-order splines, which closely resemble the Gaussian function. The important point is that the complexity of the algorithm remains the same for higher-order splines. The idea can also be extended to higher dimensions using the tensor-product of B-splines. In this case, the reduction in the complexity is even more significant. In d-dimensions, the complexity can be reduced from $O(W^d)$ to $O(1)$ using tensor-product B-splines.

1.3.1   Gaussian-like box splines

The drawback with tensor-product B-splines is that they lack spatial adaptivity. While it is possible to control their size and elongation along the image axes, it is difficult to control their elongation along arbitrary orientations.

In this thesis, we show how this can be overcome in 2-dimensions using a larger class of bivariate splines called *box splines* [6]. These are derived from B-splines, and include the tensor-product B-splines as special instances. We show that the box splines have better spatial adaptivity in the sense that we can independently control their size, elongation, and orientation. We can do so simply by controlling the width of the constituent B-splines. Importantly, as in the case of the tensor-product B-splines, we can derive fast algorithms for averaging the image using $O(1)$ computations per pixel, irrespective of the size of the box spline. The focus of the thesis is on *space-variant* filtering in which the kernel used to filter the image is varied from pixel to pixel. The overall transformation thus cannot be expressed as a convolution in this case. Nevertheless, we demonstrate that it is possible to formulate an algorithm which realizes the transformation using a constant number of operations per pixel.

1.4   Contributions

The contributions of this thesis can be grouped into the following categories.

§ *Theory*

(A) **Towards multiresolution Gabor atoms: Complexification via the Hilbert transform**.

- We provide a rigorous understanding of why the Hilbert transform goes well with wavelets. We show that at the heart of this is the characteristic vanishing-moment property of wavelets. We formulate certain basic theorems concerning the localization, smoothness, and the number of vanishing moments of the Hilbert transform of a wavelet.

- We show how the fundamental invariances of the Hilbert transform can be used to seamlessly integrate it into the multiresolution framework of wavelets. By applying the B-spline factorization theorem, we formulate a recipe for constructing a pair of wavelet bases that form Hilbert transform pairs. In particular, we are able to identify particular spline wavelets and their Hilbert transforms. By taking their complex combination, we get analytic wavelets of varying order. We show that the analytic wavelet converges to a Gabor function as its order get large.

- We identify the complete family of transforms which share the fundamental invariances of the Hilbert transform. Based on this family of transforms and

its properties, we provide an amplitude-phase interpretation of the signal representation associated with our Gabor-like wavelet transform. This provides a shiftable characterization of the transform.

(B) Beyond tensor-product B-splines for Gaussian filtering.

- We introduce a particular bivariate box spline termed the *radially-uniform box spline*. As an application of the Central Limit Theorem, we show that it converges to a Gaussian as its order gets large. For a fixed order, we show how the parameters of the box spline can be tuned to approximate a fixed anisotropic Gaussian.

§*Fast Algorithms*

(A) Coherent multiscale analysis and reconstitution using Gabor atoms.

- We provide a fast numerical implementation of the Gabor-like wavelet transform for finite periodic data in one and two dimensions.

(B) Algorithms for space-variant filtering.

- We propose a fast algorithm for space-variant image filtering using elliptical radially-uniform box splines of various shapes and sizes. The complexity of the algorithm is $O(1)$ per pixel (constant time) in the sense that it requires a fixed number of computations per pixel irrespective of the shape and size of the box splines.
- We develop an efficient root-finding algorithm for controlling the anisotropy of the *four-directional box spline*.
- We develop two fast algorithms for space-variant filtering, one modeled on anisotropic diffusion and the other on the bilateral filter.
- We show how the implementation of the non-linear bilateral filter can be accelerated using a particular class of *shiftable* range kernels.

§*Specific Applications*

The theory developed in this thesis was originally motivated by the following applications.

(A) Application of a near translation-invariant wavelet pyramid.

- We develop a coarse-to-fine stereo-matching algorithm by combining the dynamic programming algorithm with the sub-sampled Gabor-like wavelet pyramid. The crucial property that is exploited by the algorithm is the near translation-invariance of the amplitude of the Gabor-like wavelet pyramid.

(B) **Processing and analysis of biological images**.

- We develop some efficient techniques for edge-preserving smoothing of biological images using box spline filters.

- We develop a fast and accurate template-matching algorithm for detecting cells in fluorescence images using a continuously-scalable box spline detector. The algorithm returns an estimate of the number of cells along with their centers and radii. It outperforms standard convolution-based template-matching algorithms in terms of speed and accuracy.

1.5   Organization and summary

The rest of the thesis is organized as follows. In Chapter 2, we study the mathematical properties of the Hilbert transform. We introduce basic definitions and theorems which are freely used in the subsequent Chapters. We provide a self-contained exposition of some of the established results concerning the local and global behavior of transform. In particular, we study the existence, decay, and smoothness properties of the transform, its interaction with harmonics, and its behavior on the space of finite-energy signals. We go on to introduce a complete family of transforms, the so-called fractional Hilbert transforms, that share the fundamental invariances of the Hilbert transform. We also study a particular directional extension of these transforms to two and higher dimensions

In Chapter 3, we provide precise arguments as to why the Hilbert transform of a wavelet is again a wavelet. In particular, we give sharp estimates of the localization, vanishing moments, and smoothness of the transformed wavelet. We also explain why it is possible to construct a pair of (biorthogonal) wavelet bases of $\mathbf{L}^2(\mathbf{R})$, where the wavelets in one system are the Hilbert transform of those in the other system.

The problem of constructing Hilbert transform pairs of wavelet bases of $\mathbf{L}^2(\mathbf{R})$ is studied in Chapter 4. The focus is on how we can address this problem within the framework of multiresolution approximation of Mallat and Meyer. We show that the construction can be realized using a pair of multiresolutions generated by two "matched" scaling functions having identical approximation order. Inspired by the wavelet construction of Kingsbury et al. [24, 26], we extend this scheme to higher dimensions to realize two systems of wavelets that are related through the directional Hilbert transform.

Next, using the framework developed in the Chapter 4, we identify two spline multiresolutions where the wavelet bases are related through the Hilbert transform. In effect, this gives us a family of analytic spline wavelets which are continuously indexed by their order. The remarkable property is that these analytic spline wavelets converge to a Gabor function as their degree gets large. This gives us a finite-order approximation of the Gabor function, which is known to be optimally localized in the joint space-frequency domain. In higher dimensions, we construct a

family of complex spline wavelets that resemble the directional Gabor functions proposed by Daugman [3]. We also present an efficient numerical implementation of the associated complex wavelet transform using the fast Fourier transform

In Chapter 6, we provide an interpretation of the amplitude and phase information obtained from the complex Gabor-like transform. This interpretation is particularly relevant when we reconstruct the signal from the redundant wavelet coefficients, given that there exists several non-unique ways of doing so.

State-of-the-art algorithms used for dense stereo-matching use global optimization on graphs. Graph-based algorithms, however, tend to be rather slow when the size of the image and the range of the disparities are large. Image pyramiding provides an attractive means of accelerating such algorithms. In Chapter 7, we propose a "coarse-to-fine" algorithm that does narrow-band dynamic programming on the Gabor-like wavelet pyramid. The crucial feature of our pyramid is that it provides near translation-invariance at the cost of moderate redundancy. We show that a significant reduction of the computation time is obtained in comparison to the standard dynamic programming algorithm.

In Chapter 8, we demonstrate that it is possible to filter an image with a Gaussian-like elliptic window of varying size, elongation and orientation using a fixed number of computations per pixel. The filtering algorithm is based on a family of compactly supported piecewise-polynomials, and is realized using pre-integration and local finite-differences. Based on this algorithm, we develop two signal-adaptive denoising schemes in Chapter 9. The first of these is modeled on anisotropic Gaussian diffusion [31]. To approximate the anisotropic Gaussian, we use the four-directional box spline. The spatial-adaptivity in this case is in terms of the size, elongation, and orientation of the box splines. The other denoising scheme is based on a space-variant version of the Gaussian bilateral filter [32]. The spatial variability in this case is in terms of the size of the spatial Gaussian filter. We develop a fast algorithm for implementing this filter by approximating the spatial filter using the four-directional box spline, and the range filter (locally) by appropriate polynomials.

Finally, in Chapter 10, we propose a fast template-matching algorithm for the detection of cells in fluorescence images. The algorithm estimates the number of cells and their respective centers and radii. It relies on the fast computation of intensity-based correlations between the cells and a near-isotropic box spline detector. We provide experimental results on both simulated and real data to demonstrate the speed and accuracy of the algorithm.

Part I:
Optimally Localized Wavelets

Chapter 2

The Hilbert transform

Abstract — In this chapter, we study the mathematical properties of the Hilbert transform. We provide a self-contained exposition of the some of the established results concerning the local and global behavior of transform. In particular, we study the existence, decay, and smoothness properties of the transform, its interaction with harmonics, and its behavior on the space of finite-energy signals. We will require these results in the next few chapters.

Later in the chapter, we introduce a complete family of transforms, the so-called *fractional Hilbert transforms*, that share the fundamental invariances of the Hilbert transform. We also study a particular directional extension of these transforms to two and higher dimensions[1].

2.1 Introduction

INtegral transforms are commonly of the form

$$T f(x) = \int_a^b k(x,t) f(t) \, dt. \tag{2.1}$$

The function $k(x,t)$ is called the *kernel* of T. The most famous example is the Fourier transform where $k(x,t) = \exp(-jtx)$, $a = -\infty$, and $b = +\infty$.

An integral transform is said to be of the *convolution* type if $k(x,t) = K(x-t)$ for some function $K(t)$. In this case, $K(t)$ is called the kernel of the transform. When a and b are infinite (i.e., the integral is over the real line), we can express $T f(x)$ in

---

[1]Some of the results in §2.6 are from the article [33]: K. N. Chaudhury, M. Unser, "On the shiftability of dual-tree complex wavelet transforms," IEEE Transactions on Signal Processing, vol. 58, no. 1, pp. 221-232, January 2010.

either of the following forms:

$$\int_{-\infty}^{\infty} K(x-t)f(t)\,dt, \quad \text{or} \quad \int_{-\infty}^{\infty} f(x-t)K(t)\,dt.$$

A *singular* integral transform is an integral transform in which $k(x,t)$ has a singularity along the diagonal $x=t$ [34]. Under additional constraints on the kernel, it is possible to contain the singular behavior by appropriately extending definition (2.1).

In one-dimension, the archetypal singular integral transform of the convolution type is the Hilbert transform [35, 34, 36]. This is defined by

$$\mathcal{H}f(x) = \frac{1}{\pi} \lim_{\varepsilon \longrightarrow 0} \int_{|t|>\varepsilon} f(x-t)\,\frac{dt}{t}. \tag{2.2}$$

The kernel in this case is $1/\pi x$, which has a singularity at the origin. To contain the singularity, an infinitesimally small truncation around the origin is used in (2.2).

It is not immediately clear that $\mathcal{H}f(x)$ is well-defined even for nice functions. The kernel fails to be integrable both at the origin (due to the singularity) and at infinity (due to the slow decay). Thus, it is not at all obvious that the *truncated Hilbert transform*

$$\mathcal{H}_\varepsilon f(x) = \frac{1}{\pi} \int_{|t|>\varepsilon} f(x-t)\,\frac{dt}{t}. \tag{2.3}$$

converges for $\varepsilon > 0$, and that the sequence of such integrals tends to a limit as ε gets infinitesimally small. In fact, the non-trivial task in the study of the Hilbert transform is the specification of the class of functions for which (2.2) can be given a precise meaning, either pointwise or in the norm sense. In particular, we will focus on the following behaviors.

(1) **Local behavior**. The concern here is that the integral in (2.3) must be absolutely convergent for all $\varepsilon > 0$, and that $\mathcal{H}_\varepsilon f(x)$ must converge as ε goes to zero for every x. This provides a pointwise specification of $\mathcal{H}f(x)$. In this case, $f(x)$ is required to be sufficiently "nice"; that is, it must have suitable smoothness and decay properties.

(2) **Global behavior**. Here one is given a function with the single global constraint

$$\int |f(x)|^p\,dx < \infty$$

for some $1 \le p < \infty$, and is required to give a precise meaning to (2.2). Unlike the previous case, one has no explicit information about the smoothness and decay of the function. In particular, $f(x)$ could be very rough and of slow decay.

A standard way to approach this problem is to consider a dense subclass **D** of $\mathbf{L}^p$, and approximate the original function using a sequence (f_n) from **D** [36, 35]. The main point is that the functions in **D** are chosen to be of suitable smoothness and

decay, so that $\mathscr{H} f_n(x)$ is well-defined and is again in $\mathbf{L}^p$. In particular, one tries to show that $\mathscr{H}$ maps $\mathbf{D}$ continuously into $\mathbf{L}^p$, that is,

$$\|\mathscr{H} g\|_p \leq C \|g\|_p \qquad (g \in \mathbf{D}). \tag{2.4}$$

Using density and continuity, one is then able to define $\mathscr{H} f(x)$ as the (unique) limit of the sequence $\mathscr{H} f_n(x)$. Of course, $\mathscr{H} f(x)$ is defined only almost everywhere in this case.

We first address the local behavior along with the decay and smoothness properties of the transform. The global behavior is discussed next.

2.2   Local behavior

2.2.1   Existence

The Hilbert transform does not go well with rough functions. This can be readily appreciated by looking at the transform of the discontinuous function in Figure 2.2. In this case, the transform explodes in the vicinity of the points of discontinuity, and is in fact not even well-defined at these points. Therefore, the natural question is: *Under what assumptions on $f(x)$ is $\mathscr{H} f(x)$ well-defined?*

Of course, we would want to have as little assumptions as possible. The basic result in this direction is the following [36].

Proposition 1 (Existence). *Suppose that $f(x)$ is Lipschitz continuous and integrable. Then $\mathscr{H} f(x)$ is well-defined and bounded.*

In particular, this holds true if $f(x)$ is continuously differentiable and of compact support, since
$$|f(x) - f(y)| \leq \|f'\|_\infty |x - y|$$
by the mean value theorem. In this case, $\|f\|_{\mathrm{Lip}} \leq \|f'\|_\infty$.

Proof. Note that the integrand in (2.2) is the product of the bounded function $1/t$ (on $|t| > \varepsilon$) and the integrable function $f(t)$. Therefore, the integral is absolutely convergent for all $\varepsilon > 0$. All we need to show is that the integral remains convergent as $\varepsilon \longrightarrow 0$. To this end, we split the integral, and use the anti-symmetry of $1/t$ to write

$$\mathscr{H} f(x) = \frac{1}{\pi} \lim_{\varepsilon \to 0} \int_{\varepsilon < |t| < 1} f(x - t) \frac{dt}{t} + \frac{1}{\pi} \int_{|t| \geq 1} f(x - t) \frac{dt}{t}$$

$$= \frac{1}{\pi} \lim_{\varepsilon \to 0} \int K(\varepsilon, t) \left[\frac{f(x - t) - f(x)}{t} \right] dt + \frac{1}{\pi} \int_{|t| \geq 1} f(x - t) \frac{dt}{t}.$$

where $K(\varepsilon, t) = 1$ when $\varepsilon < |t| < 1$ and zero otherwise.
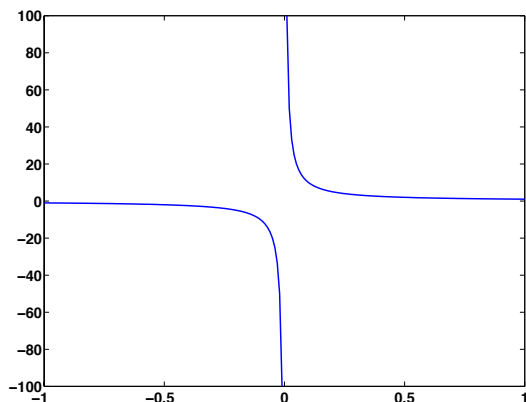
Figure 2.1: The convolution kernel of the Hilbert transform, $1/\pi x$. It has a singularity at the origin and its tails decay slowly. The former pathology can be overcome provided that the function on which the transform is applied is sufficiently smooth, while the slow decay can be overcome if the signal is of compact support, or at least, of sufficient decay.

Clearly, the second integral is convergent. On the other hand, since $f(x)$ is Lipschitz,

$$K(\varepsilon,t)\left|\frac{f(x-t)-f(x)}{t}\right| \leq \|f\|_{\mathrm{Lip}}$$

for t and for all $\varepsilon > 0$. Therefore, by dominated convergence,

$$\lim_{\varepsilon \to 0} \int K(\varepsilon,t)\left|\frac{f(x-t)-f(x)}{t}\right| dt \leq \|f\|_{\mathrm{Lip}}.$$

This shows that $\mathcal{H}f(x)$ is well-defined for every x. Note that, we also get the bound

$$|\mathcal{H}f(x)| \leq \frac{1}{\pi}\left(2\|f\|_{\mathrm{Lip}}+\|f\|_1\right). \tag{2.5}$$

This establishes the proposition. $\square$

2.2.2   Decay

Having resolved the issue of existence, our next question is the following: *How fast does $\mathcal{H}f(x)$ decay?*

A good guess is that it decays as fast as the kernel, namely, as $1/|x|$. This is indeed the case, at least if $f(x)$ is of compact support. The main idea is that if $f(x)$ is of
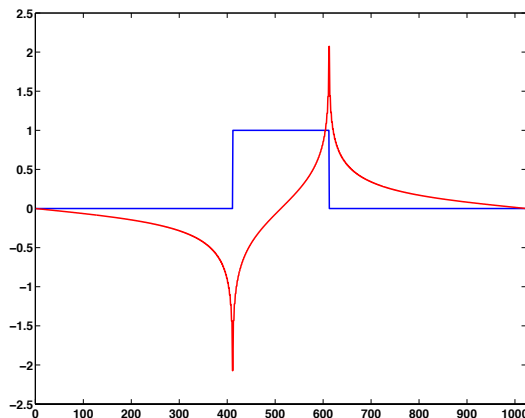
Figure 2.2: The Hilbert transform of the discontinuous $\text{rect}(x)$ function (shown in BLUE) and its transform (in RED). The transformed function explodes in the vicinity of the discontinuities and is, in fact, not even defined at the points of discontinuity. It also has a slow decay.

compact support, then for all points outside the support of $f(x)$, we can avoid the singularity and simply write (2.2) as

$$\mathscr{H}f(x) = \frac{1}{\pi} \int \frac{f(t)}{x-t}\,dt.$$

Proposition 2 (Decay). *Suppose that $f(x)$ is Lipschitz continuous and of compact support. Then $\mathscr{H}f(x)$ is well-defined and decays as $1/|x|$.*

Proof. Clearly, the transform is well-defined. Let $f(x)$ be supported within $[-T, T]$. Then, for $|x| > 2T$,

$$|\mathscr{H}f(x)| \leq \frac{1}{\pi} \int_{-T}^{T} \frac{|f(t)|}{|x-t|}\,dt < \frac{C}{|x|} \int_{-T}^{T} |f(t)|\,dt = \frac{2CT}{|x|}\|f\|_{\infty}$$

since $|x-t| > |x|/2$ when $|x| > 2T$ and $|t| \leq T$. Thus, $\mathscr{H}f(x)$ decays as $1/|x|$ for all large x. $\qquad\qquad\square$

We will show in a Chapter 3 that by appropriately modifying the assumptions, we can extend the above result to functions of non-compact support.

2.2.3 Smoothness

The kernel of the Hilbert transform is not continuous at the origin. However, $\mathscr{H}f(x)$ turns out to be continuous if $f(x)$ is sufficiently smooth.

Proposition 3 (Continuity). *Suppose that $f'(x)$ is bounded and Lipschitz continuous, and that $f(x)$ is integrable. Then $\mathcal{H}f(x)$ is uniformly continuous.*

In particular, if $f(x)$ is a compactly supported and $f'(x)$ is Lipschitz, then $\mathcal{H}f(x)$ is uniformly continuous. The proof is provided in Appendix A.

The unsatisfactory fact about this result, however, is that the smoothness of the function and its transform are not comparable—the transform is less smooth than the original function. Moreover, extending the result to higher orders of smoothness is not straightforward. We will provide a more satisfactory characterization of the smoothness using global properties of the transform in the sequel.

2.3   Global behavior

Having addressed the local behavior, namely, the pointwise specification, the decay, and the smoothness of the transform, we now turn to the global behavior. Our main concern is the following: *Is the Hilbert transform of an $\mathbf{L}^p$-function again in $\mathbf{L}^p$? If so, is it bounded on $\mathbf{L}^p$?*

The answer is in the affirmative when $1 < p < \infty$ [34, 35]. We discuss the case $p = 2$ in detail. We will require this results frequently in the sequel. For the sake of completeness, we briefly motivate and state the results when $p \neq 2$.

2.3.1 $\mathbf{L}^1$ theory

We begin with an example. Consider the function shown in Figure 2.2,

$$f(x) = \begin{cases} 1 & \text{for } 0 \leq x \leq 1, \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, this belongs to every $\mathbf{L}^p$ space. A straightforward computation shows that

$$\mathcal{H}f(x) = \log\left(\frac{|x|}{|x-1|}\right). \tag{2.6}$$

Note that $\mathcal{H}f(x)$ is not defined at the discontinuities $x = 0$ and $x = 1$, and that it decays as $1/|x|$ at infinity. Moreover, it explodes logarithmically around the discontinuities; it is unbounded there. However, it can be shown that, for every interval I around any of the discontinuities,

$$\int_I \left| \log\left(\frac{|x|}{|x-1|}\right) \right| dx < \infty$$

To see this, note that the integrand behaves as $|\log|x||$ in the neighborhood of 0,

and as $|\log|x-1||$ in the neighborhood of 1, and that, for every $0 < \varepsilon < 1/2$,

$$\int_{|x|<\varepsilon} |\log|x|| \, dx = 2 \int_{0<x<\varepsilon} |\log x| \, dx \leq 2 \int_0^\infty t \, e^{-t} dt < \infty. \qquad (2.7)$$

However, since (2.6) behaves as $1/|x|$ at the tails, the transform fails to be in $\mathbf{L}^1$. What holds true instead is the following. Under some smoothness hypothesis, it can be shown that, for every $f \in \mathbf{L}^1$,

$$|\mathcal{H}f(x)| \leq \frac{C}{1+|x|}. \qquad (2.8)$$

This means that

$$\mu\Big(\big\{x : |\mathcal{H}f(x)| \geq \lambda\big\}\Big) \leq \frac{C}{\lambda} \qquad (2.9)$$

where $\mu(A)$ denotes the Lebesgue measure of A. Technically, we say that $\mathcal{H}f(x)$ is in weak-$\mathbf{L}^1$, the space of functions satisfying the condition[2]

$$\mu\Big(\big\{x : |f(x)| \geq \lambda\big\}\Big) \leq \frac{C}{\lambda}. \qquad (2.10)$$

The following result shows that (2.9) is true for $\mathbf{L}^1$ functions in general.

Theorem 4 ($\mathbf{L}^1$ characterization). *The Hilbert transform of an $\mathbf{L}^1$ function is defined almost everywhere but does not necessarily belong to $\mathbf{L}^1$. In fact, it belongs to weak-$\mathbf{L}^1$,*

$$\mu\Big(\big\{x : |\mathcal{H}f(x)| \geq \lambda\big\}\Big) \leq \frac{C}{\lambda} \|f\|_1 \qquad (f \in \mathbf{L}^1). \qquad (2.11)$$

The converse, however, is not true. This is exactly the case for the function in (2.6), which is in weak-$\mathbf{L}^1$ but not in $\mathbf{L}^1$. In the next chapter, we will show how this situation can be improved using additional functional properties.

The original proof of this result is due to Kolmogorov [37]. The derivation is rather involved, and we refer the readers to [35] for a proof using the Calderón-Zygmund decomposition.

Sketch of the proof. As noted in the Introduction, functions in $\mathbf{L}^1$ can be quite rough and of slow decay. In reference to Proposition 1, the existence of the transform is thus not at all obvious.

---

[2]Note that weak-$\mathbf{L}^1$ is larger than $\mathbf{L}^1$. Indeed, if $f \in \mathbf{L}^1$, then

$$\|f\|_1 \geq \int_{\{x:f(x)\geq\lambda\}} |f(x)| \, dx \geq \lambda \mu(\{x : |f(x)| \geq \lambda\}).$$

The idea is to approximate a given $f \in \mathbf{L}^1$ by a sequence (f_n) from the Schwartz class. It is clear that (2.8), and hence (2.9), holds for every $f_n(x)$. The more difficult task is to establish the estimate

$$\mu\left(\{x : |\mathscr{H} f_n(x)| \geq \lambda\}\right) \leq \frac{C}{\lambda} \|f_n\|_1.$$

Using this and the fact that $\|f_n - f\|_1 \longrightarrow 0$ as n gets large, one can readily verify that the sequence $(\mathscr{H} f_n)$ is a Cauchy sequence in measure. That is, for every $\varepsilon > 0$,

$$\lim_{m,n \longrightarrow \infty} \mu\left(\{x : |\mathscr{H} f_m(x) - \mathscr{H} f_n(x)| \geq \varepsilon\}\right) = 0.$$

Thus, the limit of this sequence is an unique measurable function, which is defined to be $\mathscr{H} f(x)$. One can verify that $\mathscr{H} f(x)$ satisfies (2.11). □

We note that a necessary condition for $\mathscr{H} f(x)$ to be in $\mathbf{L}^1$ provided that $f(x)$ is in $\mathbf{L}^1$ is that

$$\int f(x)\, dx = 0. \tag{2.12}$$

This result is due to Kober [38]. That the condition is not sufficient is seen by considering the function

$$f(x) = \begin{cases} x^{-1}(\log x)^{-2} - 2(\log 2)^{-1} & \text{for } 0 < x < 1/2 \\ 0 & \text{otherwise.} \end{cases}$$

It can be verified that $f \in \mathbf{L}^1$, and that (2.12) holds. However, it can be shown that

$$\int_{-1/2}^{0} |\mathscr{H} f(x)|\, dx = \infty$$

so that the transform cannot be in $\mathbf{L}^1$.

2.3.2 $\mathbf{L}^2$ theory

We again return to (2.6). It can be verified that

$$\int \left| \log\left(\frac{|x|}{|x-1|}\right) \right|^2 dx < \infty.$$

Indeed, since the integrand behaves as $1/|x|^2$ at infinity, the integral converges at the tails. On the other hand, using an argument similar to the one used in (2.7), it can be shown that the integral converges in the neighborhoods of 0 and 1. Thus, while the transform is not integrable, it is in $\mathbf{L}^2$. The above observation is in fact part of a more general occurrence.

Theorem 5 ($\mathbf{L}^2$ characterization)**.** *The Hilbert transform of an $\mathbf{L}^2$ function is again in $\mathbf{L}^2$. Moreover, the following holds almost everywhere*

$$\widehat{\mathscr{H}f}(\omega) = -j\operatorname{sign}(\omega)\hat{f}(\omega) \qquad (f \in \mathbf{L}^2). \tag{2.13}$$

It is clear that the main conclusion follows from (2.13) by Parseval's theorem. Moreover, we also conclude the following from (2.13).

Corollary 6 (Properties on $\mathbf{L}^2$)**.** *The Hilbert transform has the following properties:*

(i) *It is unitary, $\|\mathscr{H}f\|_2 = \|f\|_2$ for all $f \in \mathbf{L}^2$.*

(ii) *$-\mathscr{H}^2$ acts as the identity operator on $\mathbf{L}^2$. This means that $\mathscr{H}$ is invertible on $\mathbf{L}^2$, where $\mathscr{H}^{-1} = -\mathscr{H}$.*

(iii) *It is skew-adjoint, $\mathscr{H}^* = -\mathscr{H}$.*

Before proceeding to the rigorous proof, we begin with a symbolic calculation of the Fourier transform of $\mathscr{H}f(x)$. Note that, formally,

$$\mathscr{H}f(x) = f(\cdot) * \frac{1}{\pi x}.$$

Therefore, by the convolution-multiplication property of the Fourier transform,

$$\widehat{\mathscr{H}f}(\omega) = \hat{f}(\omega)\widehat{\frac{1}{\pi x}}(\omega).$$

Now, applying the differentiation-multiplication property to the identity

$$\frac{d}{d\omega}[\operatorname{sign}(\omega)] = 2\delta(\omega),$$

we get

$$\widehat{\frac{1}{\pi x}}(\omega) = -j\operatorname{sign}(\omega).$$

This gives us (2.13).

Note that, all we know in this case is that

$$\int |f(x)|^2 \, dx < \infty.$$

The main problem in calculating the Fourier transform of $\mathscr{H}f(x)$ directly from (2.2) is that we do not know a prior whether the transform exists at all in this case, and whether it is in $\mathbf{L}^1$ or $\mathbf{L}^2$. Indeed, $f(x)$ could be very irregular and of slow decay. We thus proceed as in the case of Theorem 4.

Proof of Theorem 5. We first derive (2.13) on a dense subclass of $\mathbf{L}^2$, namely, the Schwartz class $\mathbf{S}$, before uniquely extending it to entire $\mathbf{L}^2$. To this end, given some $f \in \mathbf{L}^2$, we approximate it by a sequence (f_n) from $\mathbf{S}$,

$$\|f_n - f\|_2 \longrightarrow 0 \text{ as } n \longrightarrow \infty. \tag{2.14}$$

It is clear from the discussion on the local behavior of $\mathcal{H}$ that each of the $\mathcal{H}f_n(x)$ are well-defined. In fact, one can show that $\mathcal{H}f_n(x)$ is bounded and decays as $1/|x|$ asymptotically (cf. Proposition 1 of next Chapter), so that

$$\int |\mathcal{H}f_n(x)|^2 \, dx < \infty.$$

What we exactly require is a boundedness estimate as in (4.24). To this end, we compute the Fourier transform of $\mathcal{H}f_n(x)$; this is well-defined since $\mathcal{H}f_n \in \mathbf{L}^2$. In Appendix B, we show that

$$\widehat{\mathcal{H}f_n}(\omega) = -j\,\text{sgn}(\omega)\hat{f}_n(\omega). \tag{2.15}$$

This, along with Parseval's theorem, gives

$$\|\mathcal{H}f_n\|_2 = \|\widehat{\mathcal{H}f_n}\|_2 = \|\hat{f}_n\|_2 = \|f_n\|_2. \tag{2.16}$$

From (10.1), the linearity of $\mathcal{H}$, and the fact that (f_n) is a Cauchy sequence in $\mathbf{L}^2$, it follows that

$$\|\mathcal{H}f_n - \mathcal{H}f_m\|_2 = \|f_n - f_m\|_2 \longrightarrow 0 \text{ as } m, n \longrightarrow \infty.$$

That is, $(\mathcal{H}f_n)$ is a Cauchy sequence in $\mathbf{L}^2$ norm. Since $\mathbf{L}^2$ is complete, it must converge to some function. We defined this function to be $\mathcal{H}f(x)$. Note that $\mathcal{H}f(x)$ is unique—it is independent of the particular choice of the approximation sequence.

Finally, using (2.15) and applying Parseval twice, we have

$$\widehat{\mathcal{H}f}(\omega) = \lim_{n \to \infty} \widehat{\mathcal{H}f_n}(\omega) = \lim_{n \to \infty} -j\,\text{sign}(\omega)\hat{f}_n(\omega) = -j\,\text{sign}(\omega)\hat{f}(\omega)$$

where the equality holds almost everywhere. This establishes (2.13). □

Note that the main tool that is repeatedly used in the above proof is the Parseval's theorem for $\mathbf{L}^2$ functions. Indeed, this is the single important reason why the argument used in the proof breaks down for other $\mathbf{L}^p$ spaces. However, as suggested by (2.13), $\mathcal{H}$ essentially behaves like a multiplication by $\pm j$ in some sense. Since multiplication by $\pm j$ does not change the norm, one might expect $\mathcal{H}$ to be bounded on $\mathbf{L}^p$ for other p as well. This is indeed the case.

Theorem 7 (**L**^{p} characterization)**.** *The Hilbert transform is bounded on* **L**^{p} *for* $1 < p < \infty$,

$$\|\mathscr{H} f\|_p \leq C_p \|f\|_p \qquad (f \in \mathbf{L}^p).$$

For $1 < p \leq 2$, the above result can be deduced from Theorems 4 and 5 using interpolation. This can further be extended to $2 < p < \infty$ using duality and the skew-adjoint property of $\mathscr{H}$; e.g., see [34, 36]. This result was originally derived by M. Riesz using techniques from complex analysis [39].

2.3.3 Smoothness revisited

Using (2.13) and the fact that the smoothness of a function is related to the decay of its Fourier transform, we now provide a result on the smoothness of the Hilbert transform. This generalizes Proposition 3 to higher orders of smoothness.

We recall that a function $f(x)$ is said to belong to the Sobolev space $\mathbf{W}^{2,\gamma}(\mathbf{R})$, $\gamma \geq 0$, if

$$\int (1 + |\omega|^2)^\gamma |\hat{f}(\omega)|^2 \, d\omega < \infty$$

The Sobolev embedding theorem (see Appendix C) asserts that every $f(x)$ belonging to $\mathbf{W}^{2,\gamma}(\mathbf{R})$ can be identified (almost everywhere) with a function which is n-times continuously differentiable provided that $\gamma > n + 1/2$ [14]. From (2.13), we immediately conclude the following.

Proposition 8 (Comparable smoothness)**.** *If* $f(x)$ *belongs to* $\mathbf{W}^{2,\gamma}(\mathbf{R})$, *then* $\mathscr{H} f(x)$ *also belongs* $\mathbf{W}^{2,\gamma}(\mathbf{R})$. *Moreover, if* $\gamma > n + 1/2$, *then both* $f(x)$ *and* $\mathscr{H} f(x)$ *are* n-*times continuously differentiable (almost everywhere).*

2.4 Hilbert transform of harmonics

One fundamental property of the Hilbert transform is the phase-shift action

$$\mathscr{H} [\cos(\omega_0 x)] = \sin(\omega_0 x). \tag{2.17}$$

The intuitive explanation is that if we assume (2.13), then the Fourier transform of the left hand of (2.17) is

$$-j\,\mathrm{sign}(\omega) \left[\frac{\delta(\omega - \omega_0) + \delta(\omega + \omega_0)}{2} \right] = \frac{\delta(\omega - \omega_0) - \delta(\omega + \omega_0)}{2j}.$$

But this is exactly the Fourier transform of $\sin(\omega_0 x)$.

The problem with this argument, however, is that (2.13) strictly applies for functions in $\mathbf{L}^2$, and cosines are not in this space. To make sense of (2.17), we need to

appropriately interpret the integral in (2.2), namely as

$$\lim_{\varepsilon \longrightarrow 0} \int_{\varepsilon < |t| < 1/\varepsilon} f(x - t) \, \frac{dt}{t}. \tag{2.18}$$

One can verify that this is equivalent to the integral in (2.2) for the class of functions considered earlier.

Now, when $f(x) = \cos(\omega_0 x)$, we can write (2.18) as

$$\cos(\omega_0 x) \lim_{\varepsilon \longrightarrow 0} \int_{\varepsilon < |\xi| < 1/\varepsilon} \frac{\cos \xi}{\xi} \, d\xi + \sin(\omega_0 x) \lim_{\varepsilon \longrightarrow 0} \int_{\varepsilon < |\xi| < 1/\varepsilon} \frac{\sin \xi}{\xi} \, d\xi.$$

Now it can be verified, e.g., using Cauchy's integral formula and the analytic nature of e^{jz}/z on the complex plane excluding the origin, that

$$\lim_{\varepsilon \longrightarrow 0} \int_{\varepsilon < |\xi| < 1/\varepsilon} \frac{e^{j\xi}}{\xi} \, d\xi = j\pi. \tag{2.19}$$

Putting these together, we get (2.17).

We note that the basic formula (2.19) plays an important role in the derivation of the frequency response (2.13); see Appendix B.

2.4.1   Amplitude-phase modulation

We next turn to a remarkable property of the Hilbert transform concerning localized harmonics. Let $g(x)$ be a finite-energy signal whose Fourier transform is supported within $[-\Omega, \Omega]$. Then, it can verified using (2.13) that if $\omega_0 > \Omega$,

$$\mathcal{H}[g(x)\cos(\omega_0 x)] = g(x)\sin(\omega_0 x). \tag{2.20}$$

This, in fact, is a particular instance of the theorem of Bedrosian for product functions [40]. We will derive this result in a more general setting in the sequel.

This property is widely used in amplitude and phase modulation (AM-FM) in communication systems. In the former case, the problem is one of recovering the information $a(x)$ from the transmitted AM signal $f(x) = a(x)\cos(\omega_0 x + \phi)$. The modulation frequency ω_0 is typically much higher than the frequencies of $a(x)$. From (2.20), we can recover the information using the formula

$$a(x) = \sqrt{f(x)^2 + [\mathcal{H}f(x)]^2}.$$

In contrast to AM, in which the amplitude of the carrier is varied while its frequency remains constant, FM conveys the information $m(x)$ over a carrier wave by varying its instantaneous frequency. The transmitted signal in this case is given by

$$f(x) = g(x)\cos\left(\omega_0 x + 2\pi \int_0^x m(\tau) \, d\tau\right).$$

At the receiver, the signal $m(x)$ is recovered by from the instaneous frequency (derivative of the phase) of the complex signal

$$f(x) + j\mathcal{H}f(x) = g(x)\exp\left(j\omega_0 x + 2j\pi\int_0^x m(\tau)\,d\tau\right).$$

2.5 Multidimensional extension

As an extension of the Hilbert transform in higher dimensions, we consider the *directional Hilbert transform* [35, 36]. Let $\mathbf{S}^{n-1}$ denote the unit sphere in $\mathbf{R}^n$. Then corresponding to some $\boldsymbol{u} \in \mathbf{S}^{n-1}$, we define the directional Hilbert transform by

$$\mathcal{H}_{\boldsymbol{u}}f(\boldsymbol{x}) = \frac{1}{\pi}\lim_{\varepsilon\longrightarrow 0}\int_{|t|>\varepsilon} f(\boldsymbol{x} - t\boldsymbol{u})\,\frac{dt}{t}.$$

This amounts to averaging the function at position $\boldsymbol{x} \in \mathbf{R}^n$ with the Hilbert kernel $1/\pi t$ along the affine subspace $\{\boldsymbol{x} + t\boldsymbol{u} : t \in \mathbf{R}\}$.

Using the technique used to prove Theorem 5, it can be shown that $\mathcal{H}_{\boldsymbol{u}}$ takes $\mathbf{L}^2(\mathbf{R}^n)$ into $\mathbf{L}^2(\mathbf{R}^n)$, and that

$$\widehat{\mathcal{H}_{\boldsymbol{u}}f}(\boldsymbol{\omega}) = -j\,\mathrm{sign}(\boldsymbol{\omega}^T\boldsymbol{u})\hat{f}(\boldsymbol{\omega}). \tag{2.21}$$

That is, the transformation is performed with respect to the half-spaces $\{\boldsymbol{\omega} : \boldsymbol{\omega}^T\boldsymbol{u} > 0\}$ and $\{\boldsymbol{\omega} : \boldsymbol{\omega}^T\boldsymbol{u} < 0\}$ in the Fourier domain. These play a role analogous to that played by the half-lines $0 \leqslant \omega < \infty$ and $-\infty < \omega \leqslant 0$ in case of the Hilbert transform.

It is clear that the transform is invariant to (i.e., commutes with) translations and dilations, and is unitary. The key feature of the directional Hilbert transforms is that they extend the phase-shift action in (2.17) to plane waves:

$$\mathcal{H}_{\boldsymbol{u}}[\cos(\Omega\boldsymbol{u}^T\boldsymbol{x})] = \sin(\Omega\boldsymbol{u}^T\boldsymbol{x}). \tag{2.22}$$

Again, the proof of (2.22) is identical to that of (2.17).

2.6 The class of fractional Hilbert transforms

It is clear from (2.2) that $\mathcal{H}$ is invariant to translations,

$$\mathcal{H}[f(\cdot - s)](x) = (\mathcal{H}f)(x - s) \qquad (s \in \mathbf{R}).$$

Moreover, note that, for $\lambda > 0$,

$$\int_{|t|>\varepsilon} f(\lambda^{-1}x - t)\,\frac{dt}{t} = \int_{|t|>\varepsilon} f\left(\lambda^{-1}(x - \lambda t)\right)\frac{d(\lambda t)}{\lambda t} = \int_{|t|>\lambda^{-1}\varepsilon} f\left(\lambda^{-1}(x - t)\right)\frac{dt}{t}.$$
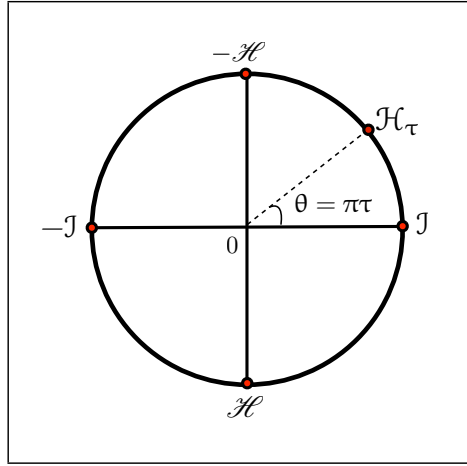
Figure 2.3: Geometrical representation of the group of fractional Hilbert transforms using the isomorphic unit-circle group S^1 (the multiplicative group of complex numbers having unit modulus). The correspondence is $\mathcal{H}_\tau \longleftrightarrow (\cos(\pi\tau), \sin(\pi\tau))$.

Letting $\varepsilon \longrightarrow 0$, we conclude that

$$(\mathcal{H}f)(\lambda^{-1}x) = \mathcal{H}[f(\lambda^{-1}\cdot)](x).$$

Thus, $\mathcal{H}$ is also invariant to dilations. Finally, note that the transform also exhibits energy-invariance—it is unitary on $\mathbf{L}^2$.

Then the natural question is: *Which other operators on* $\mathbf{L}^2$ *simultaneously exhibit the three fundamental invariances?* Of course, there is always the trivial identity operator. The following result tells us that these are essentially the only operators.

Theorem 9 (The complete family). *A unitary linear operator T on $\mathbf{L}^2(\mathbf{R})$ is invariant to translations and dilations if and only if it can be represented as*

$$T = \cos\theta \; \mathcal{I} - \sin\theta \; \mathcal{H} \tag{2.23}$$

where $\theta \in (-\pi, \pi]$ is unique.

In view of this result, we define the *fractional Hilbert transform* (fHT) corresponding to the real-valued *shift* parameter τ as

$$\mathcal{H}_\tau = \cos(\pi\tau) \; \mathcal{I} - \sin(\pi\tau) \; \mathcal{H}. \tag{2.24}$$

This definition is equivalent to the ones introduced in optics [41, 42], but differs from the ones introduced in signal processing [43, 44] up to a complex chirp.

Proof of Theorem 9. The sufficiency part of the theorem follows from the properties of the identity and Hilbert transform operator. Indeed, since $\mathscr{I}$ and $\mathscr{H}$ are invariant to both translations and dilations, so must their linear combinations. On the other hand,

$$\|Tf\|_2^2 = \cos\theta^2\|f\|_2^2 - 2\langle f, \mathscr{H}f\rangle + \sin\theta^2\|\mathscr{H}f\|_2^2 = \|f\|_2^2 - 2\langle f, \mathscr{H}f\rangle.$$

Now, using (2.13) and the symmetry $|\hat{f}(\omega)| = |\hat{f}(-\omega)|$, we have

$$\begin{aligned}
\langle f, \mathscr{H}f\rangle &= \langle \hat{f}, \widehat{\mathscr{H}f}\rangle \\
&= -j\int_{-\infty}^{0}|\hat{f}(\omega)|^2\,d\omega + j\int_{0}^{\infty}|\hat{f}(\omega)|^2\,d\omega \\
&= -j\int_{0}^{\infty}|\hat{f}(\omega)|^2\,d\omega + j\int_{0}^{\infty}|\hat{f}(\omega)|^2\,d\omega \\
&= 0.
\end{aligned}$$

This shows that T is unitary.

Next, we investigate the necessary part. It is well-known that a unitary linear operator T on $\mathbf{L}^2$ is translation invariant if and only if there exists a bounded (complex-valued) function $m(\omega)$ such that

$$\widehat{Tf}(\omega) = m(\omega)\hat{f}(\omega),$$

for all $f \in \mathbf{L}^2$ [45, Chapter 1]. This Fourier domain characterization reduces the problem to one of specifying a bounded function $m(\omega)$ such that T has the desired invariances. It is readily demonstrated that dilation-invariance is equivalent to

$$m(a\omega) = m(\omega) \qquad (a > 0). \tag{2.25}$$

Moreover, since the real and imaginary components of $m(\omega)$ must independently satisfy (2.25), one even has $m_1(a\omega) = m_1(\omega)$ and $m_2(a\omega) = m_2(\omega)$, where $m_1(\omega)$ and $m_2(\omega)$ are the real and imaginary components of $m(\omega)$.

Next, observe that the (Hermitian) symmetry requirement

$$m^*(\omega) = m(-\omega) \tag{2.26}$$

require $m_1(\omega)$ and $m_2(\omega)$ to be even and odd symmetric, respectively. One can easily verify that the only bounded functions (up to a scalar multiple) that satisfy (2.25) and (2.26) are the constant function $m_1(\omega) = 1$, and its "skewed" counterpart $m_2(\omega) = \mathrm{sign}(\omega)$. Thus, it is both necessary and sufficient that

$$m(\omega) = \gamma_1 + j\gamma_2\,\mathrm{sign}(\omega),$$

for some real γ_1 and γ_2.

Now, by Parseval's theorem,

$$\|Tf\|^2 = \frac{1}{2\pi} \int |m(\omega)|^2 |\hat{f}(\omega)|^2 \, d\omega = (\gamma_1^2 + \gamma_2^2)^2 \|f\|^2.$$

Therefore, the unitary requirement is equivalent to $\gamma_1^2 + \gamma_2^2 = 1$. In other words, $m(\omega) = \cos\theta + j\sin\theta \ \text{sign}(\omega)$ for some $\theta \in (-\pi, \pi]$, whereby we conclude that

$$T = \cos\theta \ \mathscr{I} - \sin\theta \ \mathscr{H}.$$

Finally, since $\langle \mathscr{I}f, \mathscr{H}f \rangle = 0$, $-\pi < \theta \leq \pi$ must necessarily be unique.   □

2.6.1   Properties

The characteristic features of the constituent operators $\mathscr{I}$ and $\mathscr{H}$ carry over directly to the family of fractional Hilbert transforms. In particular, we have the following properties:

- Translation invariance.

- Scale (or dilation) invariance.

- Unitarity (energy-preservation).

- Composition law $\mathscr{H}_{\tau_1} \mathscr{H}_{\tau_2} = \mathscr{H}_{\tau_1 + \tau_2}$.

- Phase-shift property $\mathscr{H}_\tau[\cos(\omega_0 x)] = \cos(\omega_0 x + \pi\tau)$.

The first three properties were described in Theorem 9. The composition law follows from the relation $-\mathscr{H}^2 = \mathscr{I}$ and basic trigonometric identities:

$$\mathscr{H}_{\tau_1} \mathscr{H}_{\tau_2} = \left[\cos(\pi\tau_1) \ \mathscr{I} - \sin(\pi\tau_1) \ \mathscr{H} \right] \left[\cos(\pi\tau_2) \ \mathscr{I} - \sin(\pi\tau_2) \ \mathscr{H} \right]$$
$$= \cos\left(\pi(\tau_1 + \tau_2)\right) \ \mathscr{I} - \sin\left(\pi(\tau_1 + \tau_2)\right) \ \mathscr{H}.$$

Finally, by (2.17),

$$\mathscr{H}_\tau[\cos(\omega_0 x + \phi)] = \cos(\pi\tau)\cos(\omega_0 x + \phi) - \sin(\pi\tau)\mathscr{H}[\cos(\omega_0 x + \phi)]$$
$$= \cos(\omega_0 x + \phi). \tag{2.27}$$

The composition law tells us that the family of fHT operators is closed with respect to composition. Moreover, as the identity $\mathscr{H}_\tau \mathscr{H}_{-\tau} = \mathscr{H}_0 = \mathscr{I}$ suggests, the inverse[3] operator is again a fHT operator, given by $\mathscr{H}_\tau^{-1} = \mathscr{H}_{-\tau}$. These closure properties can be summarized as follows.

---

[3] Note that, for a given τ, there exists infinitely many τ' such that identity $\mathscr{H}_\tau \mathscr{H}_{\tau'} = \mathscr{I}$ holds. One can however factor out the periodic structure by identifying τ_1 and τ_2 if and only if $\mathscr{H}_{\tau_1} = \mathscr{H}_{\tau_2}$, which results in the specification of an equivalence classes of fHTs. For simplicity of notation, we shall henceforth use $\mathscr{H}_\tau$ to denote both the equivalence class and its representatives.

Proposition 10 (Symmetric family)**.** *The family of fHT operators forms a commutative group on* $\mathbf{L}^2(\mathbf{R})$, $1 < p < \infty$.

In this respect, note the marked resemblance between the family of fHT operators and the commutative group of translation operators that play a fundamental role in Fourier analysis. Also note the finite subgroup $\{\mathscr{I}, \mathscr{H}, -\mathscr{I}, -\mathscr{H}\}$ of self-adjoint[4] operators (see Fig. 2.3). This is the smallest subgroup containing the in-phase and quadrature transforms.

We now record an important property concerning product functions.

Theorem 11 (Generalized Bedrosian identity)**.** *Suppose that the Fourier transform of* $f(x)$ *is supported within* $(-\omega_0, \omega_0)$, *and that the Fourier transform* $g(x)$ *vanishes inside* $(-\omega_0, \omega_0)$. *Then*

$$\mathscr{H}_\tau [f(x)g(x)] = f(x)\mathscr{H}_\tau g(x). \tag{2.28}$$

In other words, the fHT of the product of a lowpass signal and a high pass signal (with non-overlapping spectra) is entirely determined by the fHT of the high pass signal. Note that, when $\tau = -1/2$, we recover the classical result of Bedrosian for the Hilbert transform [40].

In particular, if $g(x) = \cos(\omega_0 x)$ and $\hat{f}(\omega)$ is supported within $(-\omega_0, \omega_0)$, then

$$\mathscr{H}_\tau [f(x)\cos(\omega_0 x)] = f(x)\cos(\omega_0 x + \pi\tau).$$

Proof of Theorem (2.28)*.* Note that if $f \in \mathbf{L}^2$, then

$$\widehat{\mathscr{H}_\tau f}(\omega) = m(\text{sign}(\omega))\hat{f}(\omega)$$

where $m(x) = \exp(-j\pi\tau x)$. This is immediate from (2.13).

The multiplication-convolution property of the Fourier transform shows that (2.28) is equivalent to

$$\int_{-\infty}^{\infty} \hat{f}(\omega - \xi)\hat{g}(\xi)\Big[m(\text{sgn}(\omega)) - m(\text{sgn}(\xi))\Big]\,d\xi = 0. \tag{2.29}$$

Now suppose that $\omega \geq 0$. Then (2.29) becomes

$$\int_{-\infty}^{0} \hat{f}(\omega - \xi)\hat{g}(\xi)\Big[m(1) - m(\text{sgn}(\xi))\Big]\,d\xi = 0. \tag{2.30}$$

Now, as per the assumptions, the supports of $\hat{f}(\omega - \cdot)$ and $\hat{g}(\cdot)$ are disjoint over $(-\infty, 0)$. Therefore, (2.30) holds. An identical argument shows that (2.29) holds when $\omega < 0$, which establishes the theorem. ☐

---

[4]self-adjoint up to a sign: $T^* = \pm T$ for each T in the subgroup.

2.6.2 Directional extensions

The fractional directional Hilbert transforms corresponding to direction $\boldsymbol{u} \in \mathbf{S}^{n-1}$ is defined by

$$\mathcal{H}_{\boldsymbol{u},\tau} = \cos(\pi\tau)\,\mathcal{I} - \sin(\pi\tau)\,\mathcal{H}_{\boldsymbol{u}} \qquad (\tau \in \mathbf{R}).$$

By construction, the complete family of such operators, $\{\mathcal{H}_{\boldsymbol{u},\tau} : \boldsymbol{u} \in \mathbf{S}^{n-1}, \tau \in \mathbf{R}\}$, are invariant to translations and dilations, and have the property

$$\mathcal{H}_{\boldsymbol{u},\tau}[\cos(\Omega\boldsymbol{u}^T\boldsymbol{x})] = \sin(\Omega\boldsymbol{u}^T\boldsymbol{x} + \pi\tau).$$

In particular, we have the following modulation property as an extension of (2.28).

Proposition 12. *Let the window function $\varphi(\boldsymbol{x})$ be bandlimited to the disk of radius Ω. Then*

$$\mathcal{H}_{\boldsymbol{u},\tau}\left[\varphi(\boldsymbol{x})\cos(\Omega\boldsymbol{u}^T\boldsymbol{x})\right] = \varphi(\boldsymbol{x})\sin(\Omega\boldsymbol{u}^T\boldsymbol{x} + \pi\tau). \tag{2.31}$$

Thus the transform acts only on the phase of the plane wave, while the window remains fixed.

For the sake of completeness, we characterize the complete family of unitary operators on $\mathbf{L}^2(\mathbf{R}^n)$ which are invariant to the fundamental operations of translation and scaling (dilation) on $\mathbf{R}^n$. The following result shows the resemblance of this family of operators with the fractional directional Hilbert transforms.

We call $m_T(\omega)$ the (Fourier) multiplier of an operator T on $\mathbf{L}^2(\mathbf{R}^n)$ if

$$\widehat{Tf}(\omega) = m_T(\omega)\hat{f}(\omega).$$

Theorem 13 (Invariant operators in higher dimensions). *A unitary linear operator T on $\mathbf{L}^2(\mathbf{R}^n)$ is invariant to both translations and scaling if and only if for every u belonging to the unit sphere $\mathbf{S}^{n-1}$, there exists $\tau = \tau(\boldsymbol{u})$ such that*

$$m_T(\lambda\boldsymbol{u}) = \cos(\pi\tau_{\boldsymbol{u}}) + j\sin(\pi\tau_{\boldsymbol{u}})\,\text{sign}(\lambda) \qquad (\lambda \in \mathbf{R}).$$

Note that, from (21),

$$\widehat{\mathcal{H}_{\boldsymbol{u},\tau}f}(\omega) = \left[\cos(\pi\tau_{\boldsymbol{u}}) + j\sin(\pi\tau_{\boldsymbol{u}})\,\text{sign}(\omega^T\boldsymbol{u})\right]\hat{f}(\omega).$$

Therefore, for a given $\boldsymbol{u}' \in \mathbf{S}^{n-1}$,

$$\widehat{\mathcal{H}_{\boldsymbol{u},\tau}f}(\lambda\boldsymbol{u}') = \left[\cos(\pi\tau_{\boldsymbol{u}}) \pm j\sin(\pi\tau_{\boldsymbol{u}})\,\text{sign}(\lambda)\right]\hat{f}(\omega)$$

where the sign depends on the angle between $\boldsymbol{u}'$ and $\boldsymbol{u}$.

Thus, loosely speaking, Theorem 13 tells us that every such invariant operator acts as a fractional Hilbert transform along every one-dimensional subspace in the Fourier domain. For the fractional directional Hilbert transform, the shift τ is constant over every hemisphere of $\mathbf{S}^{n-1}$.

The proof of this result is along the lines of the proof of Theorem 9, and is provided in Appendix D.

Appendix A

Consider the quantity

$$W(f) = \frac{1}{\pi} \lim_{\varepsilon \to 0} \int_{|t| > \varepsilon} f(t) \frac{dt}{t}. \qquad (2.32)$$

Note that $\mathcal{H} f(x) = W(T_x f)$, where $T_x f(t) = f(x - t)$. Since $f(x)$ is Lipschitz and integrable, $W(f)$ is well-defined by Proposition 1. Moreover, from (8.29),

$$|W(f)| \leq \frac{2}{\pi} \|f'\|_\infty + \frac{1}{\pi} \|f\|_1.$$

To establish the uniform continuity, we need to show that $\mathcal{H} f(x + h)$ approaches $\mathcal{H} f(x)$ at a uniform rate for all x as h approaches zero.

Fix some x and an arbitrary $\varepsilon > 0$. By linearity,

$$\mathcal{H} f(x + h) - \mathcal{H} f(x) = W(T_{x+h} f - T_x f).$$

Hence,

$$|\mathcal{H} f(x + h) - \mathcal{H} f(x)| \leq \frac{2}{\pi} \|T_{x+h} f' - T_x f'\|_\infty + \frac{1}{\pi} \|T_{x+h} f - T_x f\|_1.$$

Now, since the map $x \mapsto T_x f$ from $\mathbf{R}$ into $\mathbf{L}^1$ is uniformly continuous, we can find δ_1 such that

$$\|T_{x+h} f - T_x f\|_1 \leq \frac{\varepsilon}{2} \qquad (|h| < \delta_1).$$

On the other hand, $f'(x)$ being Lipschitz,

$$|T_{x+h} f'(t) - T_x f'(t)| = |f'(x + h - t) - f'(x - t)| \leq K |h|$$

In particular, if $\delta_2 = \varepsilon / 2K$, then

$$\|T_{x+h} f' - T_x f'\|_\infty \leq \frac{\varepsilon}{2} \qquad (|h| < \delta_2).$$

Letting $\delta = \min(\delta_1, \delta_2)$, we conclude that $|\mathcal{H} f(x + h) - \mathcal{H} f(x)| \leq \varepsilon$ uniformly in x provided that $|h| < \delta$. This establishes the desired result.

Appendix B

Consider the distribution W on $\mathbf{S}$ defined by

$$W(f) = \frac{1}{\pi} \lim_{\varepsilon \to 0} \int_{|t| > \varepsilon} f(t) \frac{dt}{t}.$$

Clearly, W is linear. Moreover, by modifying the derivation of (8.29), we can show that

$$|\mathscr{H}f(x)| \le \frac{1}{\pi}\left(2\|f'\|_\infty + \|xf(\cdot)\|_\infty\right). \tag{2.33}$$

This shows that the map $f \mapsto W(f)$ is continuous from $\mathbf{S}$ to $\mathbf{R}$, so that W is in $\mathbf{S}'$.

Note that the Hilbert transform of some $f \in \mathbf{S}$ is given by its convolution with W. Therefore, in the sense of distributions,

$$\widehat{\mathscr{H}f}(\omega) = \widehat{W}(\omega)\hat{f}(\omega) \qquad (f \in \mathbf{S}).$$

We now show that the distribution $\widehat{W}(\omega)$ is in fact a function.

Let $f \in \mathbf{S}$. Then, by the definition of $\widehat{W}$,

$$\begin{aligned}
\widehat{W}(f) &= W(\hat{f}) \\
&= \frac{1}{\pi}\lim_{\varepsilon\to 0}\int_{\varepsilon<|\omega|<1/\varepsilon} \hat{f}(\omega)\,\frac{d\omega}{\omega} \\
&= \frac{1}{\pi}\lim_{\varepsilon\to 0}\int_{\varepsilon<|\omega|<1/\varepsilon} \left[\int_{-\infty}^{\infty} f(x)e^{-j\omega x}\,dx\right]\frac{d\omega}{\omega}.
\end{aligned}$$

Now, clearly,

$$\int_{-\infty}^{\infty}\int_{\varepsilon<|\omega|<1/\varepsilon}\left|e^{-j\omega x}\frac{f(x)}{\omega}\right|d\omega\,dx < \infty,$$

and it can be verified that, for every $\varepsilon > 0$,

$$\left|\int_{\varepsilon<|\omega|<1/\varepsilon}\frac{e^{-j\omega x}}{\omega}\,d\omega\right| < 3.$$

Therefore, interchanging the integrals (using Fubini) and applying dominated convergence, we get

$$\begin{aligned}
\widehat{W}(f) &= \frac{1}{\pi}\lim_{\varepsilon\to 0}\int_{-\infty}^{\infty} f(x)\left[\int_{\varepsilon<|\omega|<1/\varepsilon}\frac{e^{-j\omega x}}{\omega}\,d\omega\right]dx \\
&= \frac{1}{\pi}\int_{-\infty}^{\infty} f(x)\left[\lim_{\varepsilon\to 0}\int_{\varepsilon<|\omega|<1/\varepsilon}\frac{e^{-j\omega x}}{\omega}\,d\omega\right]dx.
\end{aligned}$$

Finally, using (2.19), we obtain

$$\widehat{W}(f) = -\frac{j}{\pi}\int_{-\infty}^{\infty} f(x)\,\pi\,\mathrm{sign}(x)\,dx = \int_{-\infty}^{\infty} -j\,\mathrm{sign}(x)f(x)\,dx.$$

This shows that the Fourier transform of W is in fact the function $-j\,\mathrm{sign}(\omega)$.

Appendix C

We prove the following Sobolev embedding: If $f \in \mathbf{W}^{2,\gamma}, \gamma > n + 1/2$, then there exists a function $g \in \mathbf{C}^n$ such that $f(x) = g(x)$ almost everywhere. Here $\mathbf{C}^n$ denotes the space of functions having n continuous derivatives. On this space, we have the norm

$$\|g\|_{\infty,n} = \max_{1 \le r \le n} \sup |D^r g(x)|.$$

Note that if $g(x)$ belongs to $\mathbf{S}$, then

$$D^r g(x) = \frac{1}{2\pi} \int (j\xi)^r \hat{g}(\xi) e^{jx\xi} \, d\xi,$$

so that

$$|D^r g(x)| \le \frac{1}{2\pi} \int |\xi|^r |\hat{g}(\xi)| \, d\xi.$$

Now if $r \le n$, then by Holder's inequality,

$$|D^r g(x)| \le \left(\int \frac{|\xi|^{2r}}{(1+\xi^2)^\gamma} \, d\xi \right)^{1/2} \left(\int (1+\xi^2)^\gamma |\hat{g}(\xi)|^2 \, d\xi \right)^{1/2} = C_r \|g\|_{\mathbf{W}^{2,\gamma}}$$

where

$$C_r = \left(\int \frac{|\xi|^{2r}}{(1+\xi^2)^\gamma} \, d\xi \right)^{1/2} < \infty.$$

Note that $\|g\|_{\mathbf{W}^{2,\gamma}}$ is the Sobolev norm of $g(x)$. Setting $C = \max(C_1, \dots, C_n)$, we have the estimate

$$\|g\|_{\infty,n} \le C \|g\|_{\mathbf{W}^{2,\gamma}}.$$

We now leverage this result using the fact that $\mathbf{S}$ is dense in $\mathbf{W}^{2,\gamma}$. Consider a sequence $g_k(x)$ in $\mathbf{S}$ which converge to $f(x)$ in $\mathbf{W}^{2,\gamma}$. For each $g_k(x)$, we have

$$\|g_k\|_{\infty,n} \le C \|g_k\|_{\mathbf{W}^{2,\gamma}}.$$

We thus conclude that the sequence $g_k(x)$, which is Cauchy in $\mathbf{W}^{2,\gamma}$, is also Cauchy in $\mathbf{C}^n$. By completeness, we can then find a $g \in \mathbf{C}^n$, such that $\|g_k - g\|_{\infty,n} \longrightarrow 0$ as $k \longrightarrow \infty$.

Finally, using the fact that $g_k(x)$ converges to $f(x)$ in $\mathbf{L}^2$, and that $g_k(x)$ converges to $g(x)$ in $\mathbf{C}^n$, one can deduce that $f(x) = g(x)$ almost everywhere.

Appendix D

A continuous linear operator T on $\mathbf{L}^2(\mathbf{R}^n)$ is translation-invariant if and only if

$$\widehat{Tf}(\omega) = m_T(\omega)\hat{f}(\omega) \tag{2.34}$$

where the Hermitian symmetric multiplier $m_T(\omega)$ is such that $\|T\| = \|m_T\|_\infty$. The unitary requirement is equivalent to having $\|m_T\|_\infty = 1$.

Let $m_T(\omega) = m_1(\omega) + j\, m_2(\omega)$. Note that the Hermitian nature of $m_T(\omega)$ is equivalent to the even and odd symmetries

$$m_1(-\omega) = m_1(\omega) \quad \text{and} \quad m_2(-\omega) = -m_2(\omega). \tag{2.35}$$

On the other hand, based on (2.34) and the scaling property of the Fourier transform, we see that scale-invariance requirement is equivalent to

$$m_1(a\omega) = m_1(\omega) \quad \text{and} \quad m_2(a\omega) = m_2(\omega) \quad (a > 0). \tag{2.36}$$

Fix some $u \in \mathbf{S}^{n-1}$, and consider the function

$$\varrho_u(\lambda) = m_T(\lambda u) \qquad (\lambda \in \mathbf{R}).$$

We claim that there exists constants $c_1 = c_1(u)$ and $c_2 = c_2(u)$ such that

$$\varrho_u(\lambda) = c_1 + j\, c_2 \text{sign}(\lambda).$$

This is because (2.35) and (2.36) are equivalent to the requirements that the real part of $\varrho_u(\lambda)$ is a multiple of the constant function, and that the imaginary part of $\varrho_u(\lambda)$ is a multiple of the sign(λ) function.

Now, using (2.34) along with Parseval's theorem, we can express the unitary requirement on T in polar co-ordinates as

$$\int_0^\infty \int_{\mathbf{S}^{n-1}} |\varrho_u(\lambda)|^2 |\hat{f}(\lambda u)|^2 \, d\sigma(u) \lambda d\lambda = \int_0^\infty \int_{\mathbf{S}^{n-1}} |\hat{f}(\lambda u)|^2 \, d\sigma(u) \lambda d\lambda$$

where $d\sigma(u)$ denotes the surface measure on $\mathbf{S}^{n-1}$. The above identity obviously holds true if $|\varrho_u(\lambda)| = 1$. In fact, this condition is also necessary: Consider a radially symmetric function $f \in \mathbf{L}^2(\mathbf{R}^n)$, and observe that the modulus of $\varrho_u(\lambda)$ is independent of λ. Therefore,

$$\left(\int_{\mathbf{S}^{n-1}} |\varrho_u|^2 \, d\sigma(u) \right) \left(\int_0^\infty |\hat{f}(\lambda)|^2 \, \lambda d\lambda \right) = 2\pi \left(\int_0^\infty |\hat{f}(\lambda)|^2 \, \lambda d\lambda \right),$$

so that

$$\int_{\mathbf{S}^{n-1}} (1 - |\varrho_u|^2) \, d\sigma(u) = 0.$$

Since $|\varrho_u|^2 \leq \|m\|_\infty^2 = 1$, the integrand is non-negative, and hence $|\varrho_u| = 1$ (almost everywhere), as desired.

Therefore, we conclude that, for some $-1 \leq \tau_u \leq 1$,

$$\varrho_u(\lambda) = \cos(\pi \tau_u) + j \sin(\pi \tau_u) \, \text{sign}(\lambda).$$

Chapter 3

Hilbert transform and wavelets: From non-local to local

Abstract — A wavelet is a localized function having a prescribed number of vanishing moments. In this chapter, we provide precise arguments as to why the Hilbert transform of a wavelet is again a wavelet. In particular, we give sharp estimates of the localization (decay), oscillations (vanishing moments), and smoothness of the transformed wavelet.

We also explain why it is possible to construct a pair of (biorthogonal) wavelet bases of $\mathbf{L}^2(\mathbf{R})$, where the wavelets in one system are the Hilbert transform of the wavelets of the other system[1].

3.1 Introduction

The advantages of using Hilbert transform pairs of wavelets for signal analysis was recognized quite early on by Abry and Flandrin [46]. More recently, it has been shown that the poor translation-invariance of standard wavelet bases can be improved by considering a pair of wavelet bases, whose mother wavelets are related through the Hilbert transform [24, 27, 47]. Hilbert pairs of wavelets have also been used to improve the direction-selectivity of separable wavelet transforms in two and higher dimensions [26]

It is known for quite some time that the Hilbert transform of a wavelet is again a

---

[1]The results in this chapter are from the draft: K. N. Chaudhury, M. Unser, "On the Hilbert transform of wavelets," IEEE Transactions on Signal Processing, accepted.

wavelet. In this chapter, we are concerned with the precise understanding of the sense in which this holds true. In particular, we formulate certain basic theorems concerning the localization, smoothness, and the number of vanishing moments of the Hilbert transform of a wavelet. Our main objective was to provide self-contained and straightforward proofs of these results along with some concrete examples.

3.2   Hilbert transform and (wavelet) bases of L^2

The fundamental reasons why the Hilbert transform (in general, the fractional Hilbert transforms) can be seamlessly integrated into the multiresolution framework of wavelets are its scale and translation invariance, and its invertibility and unitarity.

The invariances of scale and translation provide coherence. The Hilbert transform of a wavelet basis generated from the mother wavelet $\psi(x)$ is simply the wavelet basis generated from the mother wavelet $\mathcal{H}\psi(x)$.

We recall from Chapter 2 that $\mathcal{H}^{-1} = -\mathcal{H}$, and that

$$\|\mathcal{H}f\|_2 = \|f\|_2 \qquad (f \in L^2). \tag{3.1}$$

This means that both $\mathcal{H}$ and its inverse are continuous mappings of L^2 onto L^2. Technically, such a map or transformation is called a *topological isomorphism* [5]. It is straightforward to verify that if T is a topological isomorphism, then

$$(f_n)_{n\in\mathbf{Z}} \text{ an orthonormal basis} \iff (Tf_n)_{n\in\mathbf{Z}} \text{ an orthonormal basis.}$$

More generally, this is true for any Riesz basis of L^2. We recall that $(f_n)_{n\in\mathbf{Z}}$ is a Riesz basis if and only if, for *every* orthonormal basis $(e_n)_{n\in\mathbf{Z}}$ of L^2,

$$f_n(x) = Te_n(x) \qquad (n \in \mathbf{Z})$$

where T is a topological isomorphism. Since the composition of topological isomorphisms is again a topological isomorphism, we conclude that

$$(f_n)_{n\in\mathbf{Z}} \text{ Riesz basis of } L^2 \iff (\mathcal{H}f_n)_{n\in\mathbf{Z}} \text{ Riesz basis of } L^2. \tag{3.2}$$

The crucial fact in connection with wavelet bases of L^2 is that the Hilbert transform preserves biorthogonal bases of L^2. We recall that the system of functions,

$$(f_n)_{n\in\mathbf{Z}} \quad \text{and} \quad (\tilde{f}_m)_{m\in\mathbf{Z}},$$

together form a biorthogonal Riesz basis of L^2 if and only if they individually form Riesz bases of L^2, and satisfy the duality criterion

$$\langle f_n, \tilde{f}_m \rangle = \delta[n-m] \qquad (m, n \in \mathbf{Z}).$$

Now, it is clear that the each of the system of functions

$$(\mathcal{H}f_n)_{n\in\mathbf{Z}} \quad \text{and} \quad (\mathcal{H}\tilde{f}_m)_{m\in\mathbf{Z}} \tag{3.3}$$
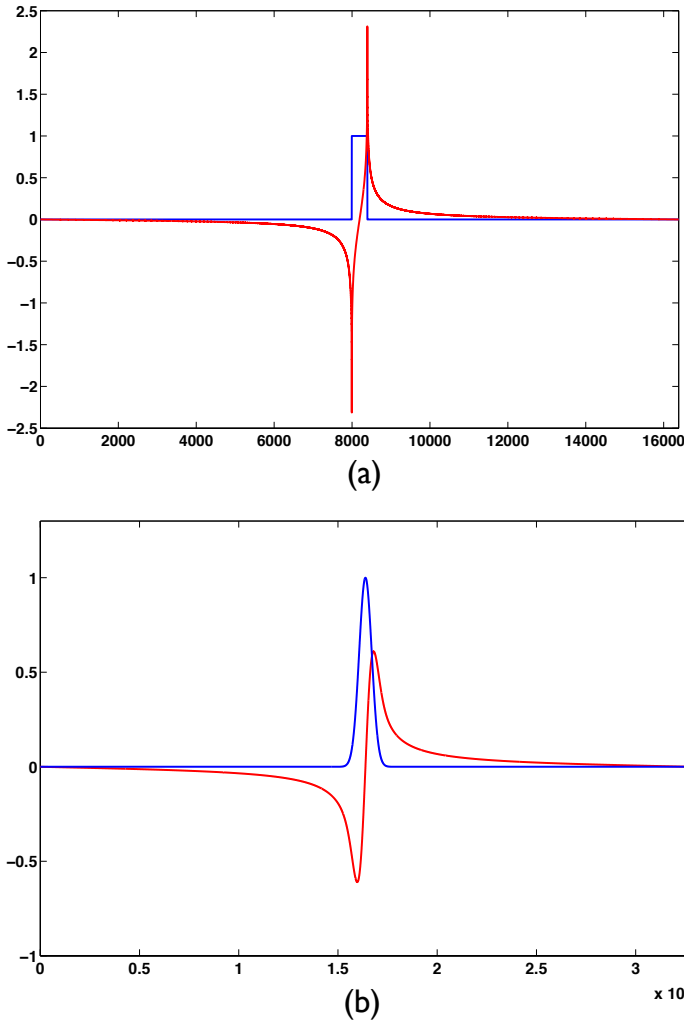
Figure 3.1: Scaling functions and their Hilbert transforms: (*a*) The discontinuous Haar scaling function (BLUE) and its transform (RED), (*b*) The smooth cubic B-spline (BLUE) and its transform (RED). In either case, the transformed function is "broken-up" and, as a consequence, loses its approximation property. In particular, the transform no longer exhibits the partition-of-unity property, which is characteristic of scaling functions. Also, note the slow decay of the transform, particular for the smooth spline function. In fact, both the transforms decay as $1/|x|$—the smoothness of the original function has no effect on the decay of the transform.

form a Riesz basis of $\mathbf{L}^2$. Moreover, it can be shown using (3.1) that

$$\langle \mathcal{H} f_n, \mathcal{H} \tilde{f}_m \rangle = \langle f_n, \tilde{f}_m \rangle = \delta[n-m].$$

Therefore the pairs in (3.3) also form a biorthogonal basis of $\mathbf{L}^2$.

3.3   Interaction with oscillating waveforms

Our main observation is that the Hilbert transform goes well with oscillatory patterns, and wavelets in particular. The archetypal relation in this regard is its action on pure sinusoids:

$$\mathcal{H}[\cos(\omega_0 x)] = \sin(\omega_0 x).$$

That is, it tends to preserve oscillations. The nature of the interaction with localized oscillations is suggested by the relation

$$\mathcal{H}[\varphi(x)\cos(\omega_0 x)] = \varphi(x)\sin(\omega_0 x) \tag{3.4}$$

which holds true if the localization window $\varphi(x)$ is bandlimited to $(-\omega_0, \omega_0)$; see Chapter 2. The crucial observation is that the transformed function is again smooth (in fact, infinitely differentiable) and oscillatory, and importantly, has the same localization as the input.

We will introduce the family of B-spline wavelets in Chapter 5. Their remarkable property is that they converge to functions of the form $\varphi(x)\cos(\omega_0 x + \phi)$ with the increase in the order of the spline. We will show that the Hilbert transform has comparable localization, smoothness, and vanishing moments as the original wavelet for sufficiently large orders; see Figure 3.2. We will also show that transformed wavelet in fact approaches the function $\varphi(x)\sin(\omega_0 x + \phi)$ as the order increases, which is consistent with (6.7). More generally, since wavelets with sufficient smoothness and vanishing moments can be made to closely approximate the form in (6.7), we could in fact arrive at a similar conclusion for a larger class of wavelets.

Using the above instances as guidelines, we attempt to answer the following fundamental questions in the sequel:

- *Why does the Hilbert transform of a wavelet decay faster than $1/|x|$, the decay of the Hilbert kernel?*

- *How good is the localization of the transformed wavelet? How many vanishing moments does it have?*

3.4   A basic result

As noted in Chapter 2, the Hilbert transform goes well only with smooth functions. This can be readily appreciated by looking at the transform of the discontinuous

Haar wavelet in Figure 3.2. In this case, the transform "blows-up" in the vicinity of the discontinuities, and is not even well-defined at these points. It was explained in Proposition 1 of Chapter 2 how this problem can be fixed, namely, by adding some smoothness constraint. We now present the same result under somewhat different hypotheses, which will be used as a basis for deriving further results.

For convenience, we introduce the mixed norm[2] $\|f\|_{1,\infty} = \|f\|_1 + \|f'\|_\infty$ which measures both the local smoothness and the global size of $f(x)$.

Proposition 14 (Basic estimate). *Let $f(x)$ be a differentiable function such that both $\|f\|_{1,\infty}$ and $\|xf(\cdot)\|_{1,\infty}$ are finite. Then $\mathcal{H}f(x)$ is well-defined, and*

$$|\mathcal{H}f(x)| \le \frac{C}{1+|x|}\left(\|f\|_{1,\infty} + \|xf(\cdot)\|_{1,\infty}\right) = O(|x|^{-1}). \tag{3.5}$$

Proof. Since $f'(x)$ is bounded, $f(x)$ is Lipschitz continuous. Then, by Proposition 1 of Chapter 2, we conclude that $\mathcal{H}f(x)$ is well-defined, and that

$$|\mathcal{H}f(x)| \le \frac{1}{\pi}\left(2\|f'\|_\infty + \|f\|_1\right). \tag{3.6}$$

As for the decay, note that

$$x\mathcal{H}f(x) = \frac{1}{\pi}\lim_{\varepsilon \to 0}\int_{|t|>\varepsilon}(x-t)f(x-t)\,\frac{dt}{t} + \frac{1}{\pi}\int f(t)\,dt$$

$$= \mathcal{H}g(x) + \frac{1}{\pi}\int f(t)\,dt$$

where $g(x) = xf(x)$. Since $\|g\|_{1,\infty}$ is finite, $\mathcal{H}g(x)$ is well-defined, and

$$|\mathcal{H}g(x)| \le \frac{1}{\pi}\left(2\|g'\|_\infty + \|g\|_1\right).$$

Therefore,

$$|x\mathcal{H}f(x)| \le \frac{1}{\pi}\left(2\|g'\|_\infty + \|g\|_1 + \|f\|_1\right). \tag{3.7}$$

Combining (3.6) and (3.7), we obtain (8.29). $\qquad\qquad\square$

Note that the above estimate holds true if $f(x)$ is continuously differentiable and is of compact support. This estimate also confirms the fact that although the Hilbert transform of a Schwartz function is in general not in $\mathbf{L}^1$, it does belong to $\mathbf{L}^2$. We recall that in Chapter 2, we derived this result using the theory of Fourier transforms on $\mathbf{L}^2$. The main objective there was to derive (3.1).

---

[2]It is clear that $\|f\|_{1,\infty}$ is non-negative, and satisfies the triangle-inequality and the homogeneity property of a norm. Moreover, it also separates functions, since $\|f\|_{1,\infty} = 0$ necessarily implies that $f(x) = 0$ identically.

3.5   Vanishing moments and decay

The derivation of Proposition 14 shows that the $1/|x|$ decay cannot be improved by requiring $f(x)$ to be more smooth (cf. transform of the cubic spline in Figure 3.1), or to have a better decay. However, it does suggest the following: If $\mathscr{H} g(x)$ goes to zero as $|x|$ goes to infinity (which is the case if $g(x)$ is sufficiently nice), then

$$\lim_{|x|\longrightarrow\infty} x\mathscr{H} f(x) = \frac{1}{\pi}\int f(x)\,dx.$$

In particular, if $f(x)$ has zero mean, then $x\mathscr{H} f(x)$ goes to zero at infinity. Therefore, the decay of $\mathscr{H} f(x)$ must be better than $1/|x|$ in this case. This alludes to the connection between the zero-mean condition and the improvement in decay. To make this more precise, we consider the concrete example of the Haar wavelet

$$\psi(x) = \begin{cases} +1 & \text{for } -1 \le x < 0 \\ -1 & \text{for } 0 \le x < 1. \end{cases}$$

Let $|x| > 2$. Since $\psi(x)$ has zero mean, we can write

$$\mathscr{H}\psi(x) = \frac{1}{\pi}\lim_{\varepsilon\to 0}\int_{|t-x|>\varepsilon}\psi(t)\left(\frac{1}{x-t}-\frac{1}{x}\right)dt$$

$$= \frac{1}{\pi}\lim_{\varepsilon\to 0}\int_{|t-x|>\varepsilon}\frac{t\psi(t)}{x(x-t)}\,dt.$$

Now $|x-t| \ge |x|/2$ for $|x| > 2$, and $t \in [-1,1]$. Hence,

$$|\mathscr{H}\psi(x)| \le \frac{2}{\pi|x|^2}\int_{-1}^{1}|t\psi(t)|\,dt \le \frac{1}{\pi|x|^2}.$$

Thus, while the Hilbert transform of the Haar scaling function decays only as $1/|x|$, the transform of the Haar wavelet has a better decay of $1/|x|^2$. This is clearly seen by comparing the plots in Figures 3.1 and 3.2.

We can now generalize the above observation by requiring that, for some $n \ge 1$,

$$\int x^k \psi(x)\,dx = 0 \qquad (0 \le k < n).$$

This *vanishing moment* property is in fact a characteristic of wavelets, which are often parametrized by the number n [14]. The following result explains how higher vanishing moments can contribute to the increase in the decay of the Hilbert transform. The main idea is that the kernel of the Hilbert transform effectively behaves as $1/\pi x^{n+1}$ in the presence of n vanishing moments.

We also use the augmented decay is used to estimate the number of vanishing moments of the transformed wavelet.

(a) Degree 0 (Harr wavelet)

(b) Degree 3 (Cubic spline wavelet)
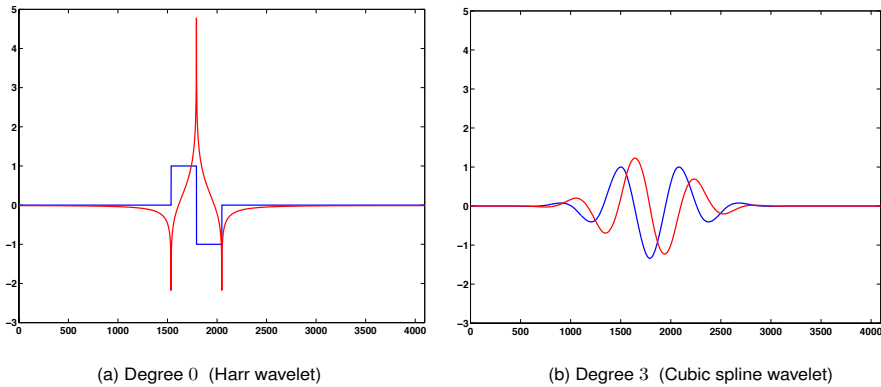
Figure 3.2: B-spline wavelets (shown in BLUE) and their Hilbert transforms (shown in RED). The wavelets are ordered (left to right) by increasing smoothness and vanishing moments; both are compactly supported. Notice how the decay of the Hilbert transform increases with the increase in vanishing moments—the transform of the cubic spline wavelet appears to have an almost identical localization. Moreover, it is as smooth as the original wavelet. It is shown in the text that, in the presence of some minimal smoothness, the Hilbert transform is as smooth and oscillating as the spline wavelet.

Theorem 15 (Decay and vanishing moments). *Let $\psi(x)$ be a differentiable wavelet having n vanishing moments. Also, assume that $\|\psi\|_{1,\infty}, \|x^{n+1}\psi(\cdot)\|_{1,\infty}$, and $\|x^n\psi(\cdot)\|_1$ are finite. Then $\mathscr{H}\psi(x)$ is well-defined, and*

$$|\mathscr{H}\psi(x)| \le \frac{C}{1+|x|^{n+1}}\left(\|\psi\|_{1,\infty} + \|x^{n+1}\psi(\cdot)\|_{1,\infty} + \|x^n\psi(\cdot)\|_1\right) = O(|x|^{-n-1}). \tag{3.8}$$

Moreover, $\mathscr{H}\psi(x)$ has n vanishing moments.

Before proceeding to the proof, we make some comments. Note that, under the assumptions on the vanishing moments, (3.8) holds true for compactly supported wavelets provided it is continuously differentiable. This in fact is the case for the cubic spline wavelet shown in Figure 3.2. More generally, (3.8) holds if $\psi(x)$ is continuously differentiable, has n vanishing moments, and satisfies the mild decay conditions

$$\psi(x) = O(1/|x|^{n+3+\varepsilon}), \quad \psi'(x) = O(1/|x|^{n+2+\varepsilon'}) \qquad (|x| \longrightarrow \infty)$$

where ε and ε' are arbitrarily small positive numbers. The significance of the above result is that by requiring $\psi(x)$ to have a large number of vanishing moments, we can effectively make $\mathscr{H}\psi(x)$ as localized as $\psi(x)$. This had been observed qualitatively early on in connection with the wavelet localization of the Radon transform [48].

Now, we show that (3.8) is sharp, by considering the special case of B-spline wavelets. It is known that if $\psi(x)$ is a B-spline wavelet of degree $n-1$, then $\mathscr{H}\psi(x)$ is again a (fractional) B-spline wavelet of the same degree, and hence has the same decay of $1/|x|^{n+1}$ [49, 50]. This is exactly what is predicted by (3.8), since $\psi(x)$ is known to have n vanishing moments.

Proof of Theorem 15. It follows from Theorem 14 that $\mathscr{H}\psi(x)$ is well-defined, and that

$$|\mathscr{H}\psi(x)| \le \frac{1}{\pi}\left(2\|\psi'\|_\infty + \|\psi\|_1\right). \tag{3.9}$$

As for the decay, fix any x away from zero, and let

$$P(t) = \frac{1}{x} + \frac{t}{x^2} + \cdots + \frac{t^{n-1}}{x^n}.$$

It is clear that

$$\int P(t)\psi(t)\,dt = 0.$$

Using this, we can write

$$\mathscr{H}\psi(x) = \frac{1}{\pi}\lim_{\varepsilon\to 0}\int_{|t-x|>\varepsilon}\frac{\psi(t)}{x-t}\,dt$$

$$= \frac{1}{\pi}\lim_{\varepsilon\to 0}\int_{|t-x|>\varepsilon}\psi(t)\Big(\frac{1}{x-t} - P(t)\Big)\,dt.$$

A simple computation shows that

$$\frac{1}{x-t} - P(t) = \frac{1}{x^{n+1}}\left(t^n + \frac{t^{n+1}}{x-t}\right),$$

so that

$$\mathcal{H}\psi(x) = \frac{1}{\pi x^{n+1}}\left[\int t^n \psi(t)\,dt + \lim_{\varepsilon \to 0}\int_{|t-x|>\varepsilon}\frac{t^{n+1}\psi(t)}{x-t}\,dt\right]$$

$$= \frac{1}{x^{n+1}}\left[\frac{1}{\pi}\int t^n \psi(t)\,dt + \mathcal{H}g(x)\right]$$

where $g(x) = x^{n+1}\psi(x)$. From Theorem 14 and the assumptions on $\psi(x)$, it follows that

$$|x^{n+1}\mathcal{H}\psi(x)| \le \frac{1}{\pi}\left(\|x^n\psi(\cdot)\|_1 + 2\|g'\|_\infty + \|g\|_1\right).$$

Combining this with (3.9), we obtain (3.8).

As for the vanishing moments of $\mathcal{H}\psi(x)$, note that, since $\psi(x)$ has n vanishing moments,

$$\int |x^k \psi(x)|\,dx < \infty \qquad (0 \le k < n).$$

One can then verify, e.g., using the dominated convergence theorem, that $\hat{\psi}(\omega)$ is n-times differentiable, and that

$$\hat{\psi}^{(k)}(0) = (-j)^n \int x^n \psi(x)\,dx \qquad (0 \le k < n). \tag{3.10}$$

Therefore, $\hat{\psi}^{(k)}(0) = 0$ for $0 \le k < n$.

Now, it can be verified (cf. Appendix A) that $\psi \in \mathbf{L}^2$. Therefore,

$$\widehat{\mathcal{H}\psi}(\omega) = -j\,\text{sign}(\omega)\hat{\psi}(\omega).$$

From this, it can be shown that $\widehat{\mathcal{H}\psi}(\omega)$ is differentiable n-times , and that $\widehat{\mathcal{H}\psi}^{(k)}(0) = 0$ for $0 \le k < n$ (cf. Appendix B). To arrive at the desired conclusion, we note that $|\mathcal{H}\psi(x)| \le C/(1+|x|^{n+1})$, so that

$$\int |x^k \mathcal{H}\psi(x)|\,dx < \infty \qquad (0 \le k < n).$$

This is sufficient to ensure that (3.10) holds for $\mathcal{H}\psi(x)$, thus completing the proof.
$\square$

We note that the specialized form of this result is well-know for the particular case of $n = 1$, that is, when the function is of zero mean. For example, along

with the classical Caldéron-Zygmund decomposition (a wavelet-like decomposition), this is used to derive certain boundedness properties of the transform on the class of integrable functions [36]. To the best of our knowledge, there is no explicit higher-order generalization of this result in the form of Theorem 15 in the harmonic analysis or signal processing literature.

Appendix A

Assume that $f(x)$ is integrable and that $f'(x)$ is bounded. We show that this implies that $f(x)$ is square-integrable.

Note that it is sufficient to show that $f(x)$ is bounded, since

$$\int |f(x)|^2 \, dx \leq \|f\|_\infty \int |f(x)| \, dx = \|f\|_\infty \|f\|_1.$$

Since $f(x)$ is continuous, a straightforward means of establishing boundedness is to show that $f(x)$ is decaying, that is, $f(x) \longrightarrow 0$ for sufficiently large x. The difficulty however is that integrability alone does not guarantee decay. Indeed, there exists integrable functions which have no decay at all. Such pathological functions can be eliminated by enforcing smoothness. What we need exactly is uniform continuity: We claim that if $f(x)$ is integrable and uniformly continuous, then it must necessarily decay.

Assume the contrary, that $f(x)$ has no decay. Then, for some $\varepsilon > 0$, we can find points $x_1, x_2, \cdots, x_n, \cdots$ such that $|f(x_n)| > \varepsilon$. By uniform continuity, there is some $\delta > 0$ such that

$$|f(x) - f(x_n)| < \frac{\varepsilon}{2} \qquad (n = 1, 2, \cdots).$$

This means that, for all $x \in (x_n - \delta, x_n + \delta)$, $|f(x)| > \varepsilon/2$. Therefore

$$\int |f(x)| \, dx \geq \sum_{n=1}^{\infty} \int_{-\delta/n}^{\delta/n} |f(x_n + y)| \, dy > \varepsilon\delta \sum_{n=1}^{\infty} \frac{1}{n} = \infty$$

which contradicts integrability.

Finally, note that $f(x)$ is indeed uniformly continuous: since $f'(x)$ is bounded, by the mean-value theorem,

$$|f(x) - f(y)| \leq \|f'\|_\infty |x - y|$$

whereby uniform continuity is immediate.

Appendix B

Let $f(x)$ be n-times differentiable, and $f^{(k)}(0) = 0$ for $0 \leq k < n$. Let

$$g(x) = \begin{cases} -cf(x) & \text{for } x > 0 \\ +cf(x) & \text{for } x < 0 \\ 0 & \text{at } x = 0 \end{cases}$$

where c is some constant. We show that $g(x)$ is n-times differentiable, and that

$$g^{(k)}(x) = \begin{cases} -cf^{(k)}(x) & \text{for } x > 0 \\ +cf^{(k)}(x) & \text{for } x < 0 \\ 0 & \text{at } x = 0. \end{cases} \tag{3.11}$$

This is conveniently done using induction. It is clear that $g(x)$ is n-times differentiable away from the origin; all we need to show that $g^{(k)}(0) = 0$ for $0 \leq k < n$. This is vacuously true for $k = 0$. Assume that (3.11) holds for some $k = m < n - 1$. Then, $g^{(m+1)}(x) = -cf^{(m+1)}(x)$ for $x > 0$, and $g^{(m+1)}(x) = +cf^{(m+1)}(x)$ for $x < 0$. At the origin,

$$g^{(m+1)}(0) = \lim_{h \to 0} \frac{g^{(m)}(h) - g^{(m)}(0)}{h}.$$

The left-sided limit is

$$g^{(m+1)}(0^-) = \lim_{h \to 0, h < 0} \frac{g^{(m)}(h) - g^{(m)}(0)}{h}$$

$$= \lim_{h \to 0, h < 0} \frac{cf^{(m)}(h) - cf^{(m)}(0)}{h},$$

which is simply $f^{(m+1)}(0^-) = 0$. Similarly, for the right-sided limit, $g^{(m+1)}(0^+) = 0$. But this means that $g^{(m+1)}(0) = 0$, which completes the induction.

Chapter 4

Hilbert transform pairs of wavelet bases

Abstract — We address the problem of constructing a pair of (biorthogonal) wavelet bases of $\mathbf{L}^2(\mathbf{R})$ where the wavelets in one system are the Hilbert transform of the wavelets of the other system. The reason why this can be realized at all is because the Hilbert transform commutes with the $ax + b$ group used in the construction of wavelet bases.

The focus of the present chapter is on how we can tackle this problem within the framework of multiresolution analysis of Mallat and Meyer. We show that the construction can be realized using a pair of multiresolutions generated by two "matched" scaling functions having identical approximation orders. Within this setting, we show that the wavelets of one multiresolution are the Hilbert transform of the wavelets of the other multiresolution provided that the corresponding wavelet filters are related through a discrete form of the continuous Hilbert transform[1].

Inspired by the construction of Kingsbury [24, 26], we extend the idea to higher dimensions to realize two systems of wavelets that are related through the directional Hilbert transform.

4.1   Introduction

The basic problem that we address in the chapter is the following:

---

[1]The present chapter is based on the article [49]: K. N. Chaudhury, M. Unser, "Construction of Hilbert transform pairs of wavelet bases and Gabor-like transforms," IEEE Transactions on Signal Processing, vol. 57, no. 9, pp. 3411-3425, September 2009.

Starting from a given wavelet basis $(\psi_{i,k})$, construct another wavelet basis $(\psi'_{i,k})$, such that

$$\psi'_{i,k}(x) = \mathscr{H}\psi_{i,k}(x) \qquad (i,k \in \mathbf{Z}). \tag{4.1}$$

We call the former the *primary* wavelet basis and the latter the *secondary* wavelet basis. We say that the primary and secondary wavelets form *Hilbert pairs*. The eventual goal is to realize a system of complex wavelets by combining these wavelets.

We recall that a wavelet basis of $\mathbf{L}^2(\mathbf{R})$ is obtained by dilations and translations of a localized oscillating waveform called the *mother wavelet*. That is, the basis consists of the functions

$$\psi_{i,k}(x) = 2^{-i/2}\psi(2^{-i}x - k).$$

where $\psi(x)$ is the mother wavelet. It was remarked in Chapter 3 that (4.1) can be realized simply by identifying a wavelet basis $(\psi'_{i,k})$ where the mother wavelet $\psi'(x)$ satisfies the condition

$$\psi'(x) = \mathscr{H}\psi(x). \tag{4.2}$$

In this chapter, $f'(x)$ should not be confused with the derivative of $f(x)$.

4.2   Multiresolution analysis

The recipe for constructing wavelet bases is given by the framework of *multiresolution approximation* proposed by Mallat and Meyer [51, 14]. We briefly recall the essential aspects of this framework.

A sequence of closed subspaces

$$\{0\} \subset \cdots \subset \mathbf{V}_1 \subset \mathbf{V}_0 \subset \mathbf{V}_{-1} \subset \cdots \subset \mathbf{L}^2(\mathbf{R})$$

is said to be a multiresolution approximation of $\mathbf{L}^2(\mathbf{R})$ provided that

(i) $f(x)$ belongs to $\mathbf{V}_i$ if and only if $f(x - 2^i k)$ is in $\mathbf{V}_i$ for every $k \in \mathbf{Z}$.

(ii) $f(x)$ belongs to $\mathbf{V}_i$ if and only if $f(2^{-1}x)$ is in $\mathbf{V}_{i+1}$ for every $i \in \mathbf{Z}$.

(iii) The only function which belongs to every $\mathbf{V}_i$ is the zero function, that is,

$$\bigcap_{i \in \mathbf{Z}} \mathbf{V}_i = \{0\}.$$

(iv) The system $(\mathbf{V}_i)_{i \in \mathbf{Z}}$ is dense in $\mathbf{L}^2$,

$$\text{closure}\left(\bigcup_{i \in \mathbf{Z}} \mathbf{V}_i\right) = \mathbf{L}^2.$$

This means that every function in $\mathbf{L}^2(\mathbf{R})$ can be approximated arbitrary closely using functions from $(\mathbf{V}_i)$.

(v) There exists $\varphi(x)$ such that the system of translates $\varphi(x-k), k \in \mathbf{Z}$, form a Riesz basis of $\mathbf{V}_0$. This provides a stable and unique representation of functions in $\mathbf{V}_0$ using coefficients from $\ell^2(\mathbf{Z})$.

The final criterion brings us to the key concept of a *scaling function*. We call $\varphi(x)$ a scaling function if there exists a sequence $h[k]$ such that

$$\frac{1}{2}\varphi\left(\frac{x}{2}\right) = \sum_{k \in \mathbf{Z}} h[k]\varphi(x-k). \tag{4.3}$$

This is called the *two-scale relation*, and $h[k]$ is referred to as the *refinement filter*. Note that conditions (i) and (ii) are satisfied if we set[2]

$$\mathbf{V}_i = \mathrm{span}\left\{\varphi(2^{-i}x - k): \ k \in \mathbf{Z}\right\}.$$

On the other hand, it can be shown that if the translates of $\varphi(x)$ form a *partition-of-unity*, that is,

$$\sum_{k \in \mathbf{Z}} \varphi(x-k) = 1, \tag{4.4}$$

then the multiresolution is dense in $\mathrm{L}^2(\mathbf{R})$, that is, (iv) is satisfied [14].

Henceforth, we will call $\varphi(x)$ a *valid* scaling function if it satisfies the Riesz basis property along with (4.3) and (4.4).

4.2.1 Wavelet basis

Let $\mathbf{W}_0$ be the complement (not necessarily orthogonal) of $\mathbf{V}_0$ in $\mathbf{V}_{-1}$. A wavelet $\psi(x)$ is defined to be the function whose translates span the space $\mathbf{W}_0$, that is,

$$\mathbf{W}_0 = \mathrm{span}\left\{\psi(x-k): k \in \mathbf{Z}\right\}.$$

In particular, $\psi \in \mathbf{V}_{-1}$. By the multiresolution property (ii), $\psi(2^{-1}x)$ belongs to $\mathbf{V}_0$. Thus, there exists a sequence $g[k]$ such that

$$\frac{1}{2}\psi\left(\frac{x}{2}\right) = \sum_{k \in \mathbf{Z}} g[k]\varphi(x-k). \tag{4.5}$$

We call $g[k]$ the *wavelet filter*.

In general, it follows from (i) and (ii) that if $\mathbf{W}_i$ is the complement (not necessarily orthogonal) of $\mathbf{V}_i$ in $\mathbf{V}_{i-1}$, then the wavelets

$$\psi_{i,k}(x) = 2^{-i/2}\psi(2^{-i}x - k)$$

---

[2]Technically, it is the closure of the span.

span the space $\mathbf{W}_i$. From (iii) and (iv), it is straightforward to verify that the system $(\psi)_{i,k}, i, k \in \mathbf{Z}$, forms a Riesz basis of $\mathbf{L}^2$. This was a brief overview of how wavelet bases are created using multiresolutions.

In the particular case of an orthonormal system, the sequences $h[k]$ and $g[k]$ are so chosen that

$$\langle \psi_{i,k}, \psi_{i',k'} \rangle = \delta[i - i']\delta[k - k'].$$

This means that the system $(\psi_{i,k})$ forms an orthonormal basis of $\mathbf{L}^2$. The corresponding wavelet representation is given by

$$f(x) = \sum_{(i,k) \in \mathbf{Z}^2} \langle f, \psi_{i,k} \rangle \psi_{i,k}(x) \qquad (f \in \mathbf{L}^2).$$

4.2.2 Biorthogonal wavelet bases

The condition of orthonormality often turns out to be rather restrictive for wavelet design. More generally, one can also have biorthogonal wavelet bases of $\mathbf{L}^2$ where distinct analysis and synthesis wavelets are used [14].

We recall that if (f_n) is a basis of $\mathbf{L}^2$, then a system (g_n) in $\mathbf{L}^2$ is said to be its *dual basis* if

$$\langle f_n, g_m \rangle = \delta[n - m] \qquad (m, n \in \mathbf{Z}). \tag{4.6}$$

It can be shown that (g_n) is unique and forms a basis of $\mathbf{L}^2$ [5]. We call the pair (f_n) and (g_n) a *biorthogonal basis* of $\mathbf{L}^2$. We represent the dual of (f_n) by $(\tilde{f}_n)$.

For constructing biorthogonal wavelet bases, we require two distinct multiresolutions

$$\{0\} \subset \cdots \subset \mathbf{V}_1 \subset \mathbf{V}_0 \subset \mathbf{V}_{-1} \subset \cdots \subset \mathbf{L}^2(\mathbf{R})$$

and

$$\{0\} \subset \cdots \subset \tilde{\mathbf{V}}_1 \subset \tilde{\mathbf{V}}_0 \subset \tilde{\mathbf{V}}_{-1} \subset \cdots \subset \mathbf{L}^2(\mathbf{R}).$$

These are generated by two distinct scaling functions $\varphi(x)$ and $\tilde{\varphi}(x)$. Let the complementary spaces be

$$\mathbf{W}_i = \mathbf{V}_{i-1} \ominus \mathbf{V}_i \quad \text{and} \quad \tilde{\mathbf{W}}_i = \tilde{\mathbf{V}}_{i-1} \ominus \tilde{\mathbf{V}}_i.$$

As before, we consider a wavelet $\psi(x)$ whose translates span $\mathbf{W}_0$, and another wavelet $\tilde{\psi}(x)$ whose translates span $\tilde{\mathbf{W}}_0$. The wavelets are specified by

$$\frac{1}{2}\psi\left(\frac{x}{2}\right) = \sum_{k \in \mathbf{Z}} g[k]\varphi(x - k) \tag{4.7}$$

and

$$\frac{1}{2}\tilde{\psi}\left(\frac{x}{2}\right) = \sum_{k \in \mathbf{Z}} \tilde{g}[k]\tilde{\varphi}(x - k). \tag{4.8}$$

The fundamental feature of the construction is that ensembles $(\psi_{i,k})$ and $(\tilde{\psi}_{i',k'})$ form a dual basis of $\mathbf{L}^2$. The corresponding wavelet representation is given by

$$f(x) = \sum_{(i,k)\in\mathbf{Z}^2} \langle f, \tilde{\psi}_{i,k} \rangle \psi_{i,k}(x).$$

The wavelets $(\psi_{i,k})$ and $(\tilde{\psi}_{i,k})$ are respectively referred to as the *synthesis* and *analysis* wavelets, or also as the *primal* and *dual* wavelets. Clearly, their roles are reversible in that

$$f(x) = \sum_{(i,k)\in\mathbf{Z}^2} \langle f, \psi_{i,k} \rangle \tilde{\psi}_{i,k}(x).$$

4.3 Hilbert transform and multiresolutions

As explained in Chapter 3, the Hilbert transform goes well with oscillatory waveforms, wavelets in particular. The difficulty with the transform in the context of multiresolution analyses is that it is incompatible with scaling functions, the building block of multiresolutions. The causes of difficulty are the following.

(i) **Poor localization**. We recall from Chapter 2 that the Hilbert transform of a function decays as $1/|x|$ in general. This results in the spatial de-localization of a scaling function, particularly, if it is of compact support. On the other hand, as shown in Chapter 2, the localization of the Hilbert transform of a wavelet can effectively be matched with that of the original wavelet if it has a large number of vanishing moments. This leads to an unbalanced situation where the scaling function and wavelet (derived from the scaling function) of the secondary multiresolution are not comparable.

(ii) **Loss of approximation property**. The other difficulty with the transform is that it "breaks-up" bump functions. The culprit in this case is the "oscillating" form of the Hilbert kernel. This results in the loss of the crucial approximation property of scaling functions. Indeed, it can be verified that both $\varphi(x)$ and $\mathcal{H}\varphi(x)$ cannot satisfy (4.4), which is essential for the multiresolution to be dense in $\mathbf{L}^2$.

Starting from a given multiresolution with associated wavelet basis $(\psi_n)_{n\in\mathbf{Z}}$, (i) and (ii) present difficulties in realizing a dual multiresolution with basis $(\mathcal{H}\psi_n)_{n\in\mathbf{Z}}$. In the sequel, we show how this can be overcome by a careful design of the dual multiresolution in which the Hilbert transform is applied only on the wavelet, and never explicitly on the scaling function.

First, we consider a particular class of scaling functions whose Hilbert transform can be well-characterized. In doing so, we introduce a discrete form of the Hilbert filter.

4.4   Fractional B-splines

The family of *fractional B-splines* introduced was introduced and studied by Blu
and Unser in [21, 22]. These are the fractional extensions of the Schoenberg's
family of polynomial B-splines [9]. We recall that the fractional B-spline $\beta_\tau^\alpha(x)$,
corresponding to a non-negative degree $\alpha \in \mathbf{R}$ and a shift $\tau \in \mathbf{R}$, is given by the
Fourier transform

$$\hat{\beta}_\tau^\alpha(\omega) = \left(\frac{1 - e^{-j\omega}}{j\omega} \right)^p \left(\frac{1 - e^{j\omega}}{-j\omega} \right)^q \tag{4.9}$$

where

$$p = \frac{1}{2}(\alpha + 1) + \tau \quad \text{and} \quad q = \frac{1}{2}(\alpha + 1) - \tau.$$

The parameters α and τ control the width and the group delay of the function,
respectively. To see the connection with the polynomial B-splines, note that when
$\tau = (\alpha + 1)/2$,

$$\hat{\beta}_\tau^\alpha(\omega) = \left(\frac{1 - e^{-j\omega}}{j\omega} \right)^{\alpha+1}.$$

This is the Fourier transform of the conventional polynomial B-spline when α is an
integer [9]. Following the notation in [22], we denote this by $\beta_+^\alpha(x)$.

4.4.1   Basic properties

The fractional B-splines do not have a compact support, except when α is an integer
(polynomial B-splines). They are bounded and have a polynomial decay. In fact, as
shown in [22, Theorem 3.1],

$$|\beta_\tau^\alpha(x)| \leq \frac{C}{1 + |x|^{\alpha+2}} \qquad (\alpha \geq 0).$$

This ensures their inclusion in all $\mathbf{L}^p$ spaces. It is clear that the modulus of (4.9) is
independent of τ; it only influences the phase term in (4.9). We will use this fact
frequently in the sequel.

The crucial property of the fractional B-splines is that they are valid scaling func-
tion. It is seen from (4.9) that the corresponding refinement filter is given by

$$H_\tau^\alpha(e^{j\omega}) = \left(\frac{1 + e^{-j\omega}}{2} \right)^p \left(\frac{1 + e^{j\omega}}{2} \right)^q. \tag{4.10}$$

The fractional B-splines are in fact intimately related with *every* valid scaling func-
tion. This is the so-called *B-spline factorization theorem*. It requires the notion of
approximation order which will be discussed in detail shortly.

4.4.2   Hilbert transform of B-splines

A direct computation of the Hilbert transform of a function is not straightforward. To compute the transform for the fractional B-splines, we take a different route which is based on the following observation.

A well-known fact in spline theory is that the derivative of a polynomial B-spline is given by the finite-difference of B-splines of lower degree. This is easily seen in the Fourier domain by noting that the multiplier for the derivative operator, $j\omega$, and that for the finite-difference operator, $1 - e^{-j\omega}$, are built into (4.9) . In particular,

$$(j\omega)\hat{\beta}^n_+(\omega) = (j\omega)\left(\frac{1 - e^{-j\omega}}{j\omega}\right)^{n+1}$$

$$= (1 - e^{-j\omega})\left(\frac{1 - e^{-j\omega}}{j\omega}\right)^n$$

$$= (1 - e^{-j\omega})\hat{\beta}^{n-1}_+(\omega).$$

This provides an algebraic explanation of the connection between the derivative and the finite-difference. This easily generalizes to the Hilbert transform for which the multiplier can be written as

$$-j\,\mathrm{sign}(\omega) = (j\omega)^{-1/2}(-j\omega)^{1/2}. \tag{4.11}$$

Indeed, when $\omega > 0$,

$$\pm j\omega = |\omega|e^{\pm\pi/2}.$$

Therefore,

$$(j\omega)^{-1/2}(-j\omega)^{1/2} = (|\omega|e^{\pi/2})^{-1/2}(|\omega|e^{-\pi/2})^{1/2} = -j.$$

On the other hand, when $\omega < 0$, the right side of (4.11) equals j.

Note that the right side of (4.11) is exactly the factor that appears in the denominator of (4.9). Proceeding as before, we have

$$(j\omega)^{-1/2}(-j\omega)^{1/2}\hat{\beta}^\alpha_\tau(\omega) = (j\omega)^{-1/2}(-j\omega)^{1/2}\left(\frac{1 - e^{-j\omega}}{j\omega}\right)^p\left(\frac{1 - e^{j\omega}}{-j\omega}\right)^q$$

$$= (1 - e^{-j\omega})^{-1/2}(1 - e^{j\omega})^{1/2}\hat{\beta}^\alpha_{\tau+1/2}(\omega). \tag{4.12}$$

Let us define

$$D(e^{j\omega}) = \begin{cases} (1 - e^{-j\omega})^{-1/2}(1 - e^{j\omega})^{1/2} & \text{for } \omega \in (-\pi, \pi)\backslash\{0\} \\ 0 & \text{when } \omega = 0. \end{cases} \tag{4.13}$$

A simple computation shows that

$$D(e^{j\omega}) = -j\,\mathrm{sign}(\omega)\exp(-j\omega/2)$$

By Parseval's theorem, we can defined a bounded operator Δ on $\mathbf{L}^2$ by

$$\widehat{\Delta f}(\omega) = D(e^{j\omega})\hat{f}(\omega). \tag{4.14}$$

In fact, similar to the Hilbert transform, Δ is also unitary. Moreover, since $D(e^{j\omega})$ is periodic, we can associate with it a digital filter $d[k]$ such that

$$\Delta f(x) = \sum d[k] f(x-k).$$

The inverse Fourier transform gives

$$
\begin{aligned}
d[k] &= \frac{1}{2\pi} \int_{-\pi}^{\pi} D(e^{j\omega})e^{jk\omega} \, d\omega \\
&= \frac{1}{2\pi} \int_{-\pi}^{\pi} -j\,\mathrm{sign}(\omega)e^{j(k+1/2)\omega} \, d\omega \\
&= \frac{1}{\pi(k+1/2)}.
\end{aligned}
$$

The filter $d[k]$ can be interpreted as a discrete form of the continuous Hilbert filter $1/\pi x$. It acts as an unitary operator on $\mathbf{L}^2(\mathbf{R})$ when applied to functions, and as a digital filter on $\ell^2(\mathbf{Z})$ when applied to sequences.

From (4.12) and (4.14), we have the following formula.

Proposition 16 (The key formula). *The Hilbert transform of a B-spline is given by*

$$\mathcal{H}\beta_\tau^\alpha(x) = \sum_{k \in \mathbf{Z}} d[k]\beta_{\tau+1/2}^\alpha(x-k). \tag{4.15}$$

As discussed earlier, the theoretical difficulty with the Hilbert transform stems from the fact that its frequency response has a discontinuity at the origin. This creates a discontinuity of the Fourier transform of the resultant function at the origin. In the space domain, the effect is that we have a poor $1/|x|$ decay.

The remarkable feature of (4.15) is that we have been able to express the slowly decaying transform as a linear combination of better-behaved B-splines. By doing so, we have, in effect, moved the discontinuity onto the digital filter. In the sequel, we apply this to the wavelet filter $g[k]$ where its effect is much more innocuous, since the Fourier transform of $g[k]$ vanishes in a neighborhood of the origin. This can be seen as the "discrete" counterpart of the ideas developed in Chapter 3 concerning the "continuous" Hilbert transform.

4.5   Hilbert pairs of wavelet bases

Before stating the main result, we recall a spectral factorization result for scaling functions due to Unser et al. [52]. To do so, we need the notion of approximation order (or simply order) that characterizes its approximation power.

Corresponding to a scaling function $\varphi(x)$, consider the spaces

$$\mathbf{U}_a = \text{span}\left\{\varphi(a^{-1}x - k) : k \in \mathbf{Z}\right\} \qquad (a > 0).$$

When $a = 2^i$, $i \in \mathbf{Z}$, these correspond to the multiresolution subspaces of Mallat's construction.

Given some $f(x)$ in $\mathbf{L}^2$, consider its orthogonal projection $P_a f(x)$ onto $\mathbf{U}_a$. That is, $P_a f(x)$ is the "best" representation of $f(x)$ in $\mathbf{U}_a$ in the sense of minimum $\mathbf{L}^2$ error. From the density of $(\mathbf{U}_a)_{a>0}$ in $\mathbf{L}^2$, it is clear that

$$\lim_{a \to 0} \|f - P_a f\|_2 = 0.$$

The approximation order provides a characterization of this rate of decay in terms of the scale a for sufficiently regular functions. We say that $\varphi(x)$ has an approximation order γ if and only if

$$\|f - P_a f\|_2 \leq C a^\gamma \|\partial^\gamma f\|_2 \qquad (f \in \mathbf{W}^{2,\gamma}). \tag{4.16}$$

Here $\partial^\gamma f(x)$ is the (distributional) derivative of order γ, defined by

$$\widehat{\partial^\gamma f}(\omega) = (j\omega)^\gamma \hat{f}(\omega).$$

Note that $\|\partial^\gamma f\|_2$ is finite since $f \in \mathbf{W}^{2,\gamma}$.

4.5.1 B-spline factorization

Akin to their polynomial counterparts, the order of fractional B-splines is entirely controlled by their degree. In particular, it can be shown that $\gamma = \alpha + 1$ [22].

A fundamental result in wavelet theory is that it is always possible to express a valid scaling function as a convolution between a fractional B-spline and a distribution. The result in [52] involves causal B-splines. This can, however, be readily extended to the more general fractional B-splines since the shift parameter τ does not influence the order of the scaling function. Note that the theorem in [52] asserts that $H(e^{j\omega})$ is the refinement filter corresponding to a valid scaling function of order $\alpha + 1$ if and only if

$$H(e^{j\omega}) = \left(\frac{1 + e^{-j\omega}}{2}\right)^{\alpha+1} Q(e^{j\omega}) \tag{4.17}$$

where $Q(e^{j\omega})$ is bounded. We recognize the first term as the refinement filter of the causal fractional B-spline. Now, we can write (4.17) as

$$H(e^{j\omega}) = \left(\frac{1 + e^{-j\omega}}{2}\right)^p \left(\frac{1 + e^{j\omega}}{2}\right)^q P(e^{j\omega}) \tag{4.18}$$

where

$$P(e^{j\omega}) = \exp\left[-j\omega\left(\frac{\alpha + 1}{2} - \tau\right)\right] Q(e^{j\omega}).$$

Note that $P(e^{j\omega})$ is again bounded. Thus, $H(e^{j\omega})$ is the refinement filter of a valid scaling function of order $\alpha + 1$ if and only if it admits a factorization as in (4.18).

By noting that the leading term in (4.18) is simply the filter in (4.10), we have the following conclusion.

Theorem 17 (B-spline factorization). *A valid scaling function $\varphi(x)$ is of order $\alpha + 1$ if and only if, for some real τ,*

$$\hat{\varphi}(\omega) = \hat{\beta}_\tau^\alpha(\omega)\hat{\varphi}_0(\omega)$$

where $\hat{\varphi}_0(\omega)$ is bounded on every compact interval, and equals unity at the origin.

In the space domain, this corresponds to a well-defined convolution

$$\varphi(x) = \beta_\tau^\alpha * \varphi_0(x). \tag{4.19}$$

The technical part of the statement is that $\hat{\varphi}_0(\omega)$ does not necessarily correspond to some true function in the space domain—it is essential to interpret φ_0 as a distribution. This requires the convolution in (4.19) to be interpreted appropriately.

A deep implication of Theorem 17 is that all desirable features of the scaling function, including its approximation property, can be explained using the properties of the constituent B-splines [52].

4.5.2   The basic construction

We now propose a method for realizing the Hilbert transform of a given wavelet within the multiresolution framework. Let the primary wavelet be given by

$$\frac{1}{2}\psi\left(\frac{x}{2}\right) = \sum g[k]\varphi(x - k).$$

We need to specify a secondary scaling function $\varphi'(x)$ and a wavelet filter $g'[k]$ such that the corresponding wavelet

$$\frac{1}{2}\psi'\left(\frac{x}{2}\right) = \sum g'[k]\varphi'(x - k) \tag{4.20}$$

such that

$$\psi'(x) = \mathscr{H}\psi(x). \tag{4.21}$$

Since the concerned systems are linear, a trivial means of realizing (4.21) would be by setting

$$\varphi'(x) = \mathscr{H}\varphi(x) \quad \text{and} \quad g'[k] = g[k].$$

This, however, leads to a paradoxical situation. Suppose that $\varphi(x)$ and $\psi(x)$ decay as $1/|x|^3$, and that $\psi(x)$ has two vanishing moments. Then $\varphi'(x)$ will have a decay of $1/|x|$, while the corresponding wavelet $\psi'(x)$ will decay as $1/|x|^3$.

The other problem is that we assume that the Hilbert transform of a scaling function is again a scaling function. As explained in §4.3, this is not true in general.

Our main observation is that (4.21) can be realized by identifying a secondary scaling function $\varphi'(x)$ whose approximation properties are comparable to that of $\varphi(x)$. This is exactly where we require the B-spline factorization theorem. Of course, we require a knowledge of the approximation order to use the theorem.

Let $\varphi(x)$ be of order $\alpha + 1$. Then by (4.19), for some τ and distribution φ_0,

$$\varphi(x) = \beta_\tau^\alpha * \varphi_0(x). \tag{4.22}$$

Consider the function

$$\varphi'(x) = \beta_{\tau+1/2}^\alpha * \varphi_0(x). \tag{4.23}$$

Clearly, $\varphi'(x)$ is a valid scaling function and has the same order as $\varphi(x)$. Let $\psi'(x)$ be any arbitrary wavelet within the multiresolution analysis of $\varphi'(x)$ given by (4.20). In this setting, the following result provides a necessary and sufficient condition for (4.21) to hold.

Theorem 18 (Hilbert pair of wavelets). *Within the above settings, the wavelets $\psi(x)$ and $\psi'(x)$ satisfy (4.21) if and only if $g'[k] = d * g[k]$.*

Proof. By linearity and continuity of $\mathcal{H}$,

$$\mathcal{H}\psi(x/2) = \sum_{k \in \mathbf{Z}} g[k] \mathcal{H}\varphi(x - k).$$

Using (4.22) and (4.15), and formally commuting the convolutions, we have

$$\mathcal{H}\varphi(x) = \left(\mathcal{H}\beta_\tau^\alpha\right) * \varphi_0(x)$$
$$= \sum_{n \in \mathbf{Z}} d[n] \beta_{\tau+1/2}^\alpha * \varphi_0(x - n).$$

Combining the above relations and applying (4.23), we get

$$\mathcal{H}\psi(x/2) = \sum_{k \in \mathbf{Z}} d * g[k] \varphi'(x - k). \tag{4.24}$$

The sufficiency of the condition $g'[k] = d * g[k]$ is now immediate from (4.20) and (4.24). Conversely, if we assume (4.21), then by (4.20) and (4.24),

$$\sum_{k \in \mathbf{Z}} g'[k] \varphi'(x - k) = \sum_{k \in \mathbf{Z}} d * g[k] \varphi'(x - k).$$

Since the translates of $\varphi'(x)$ form a Riesz basis of its span, every element in it must necessarily have a unique representation. Therefore, $g'[k] = d * g[k]$. $\qquad\square$

We make the following remarks about our construction.

(1) **System conditioning**. Note that both $\varphi(x)$ and $\varphi'(x)$ have the same Riesz bounds. This is because the autocorrelation sequences in either case, $\langle \varphi(\cdot), \varphi(\cdot - k) \rangle$ and $\langle \varphi'(\cdot), \varphi'(\cdot - k) \rangle$ are identical.

(2) **Localization**. Since the B-splines in (4.22) and (4.23) have the same decay, the same is true for $\varphi(x)$ and $\varphi'(x)$. Importantly, this means that the scaling filters of the primary and secondary multiresolution have similar localizations.

On the other hand, wavelets filters are typically required to be in ℓ^1, which gives the best localization in terms of ℓ^p spaces. In this context, the main difficulty (as with the continuous counterpart) is that the filter $d[k]$ decays only as $1/|k|$—it is not integrable. The best we can say is that it belongs to ℓ^2. This is however not sufficient to guarantee the inclusion of $d * g[k]$ in ℓ^1 even when $g[k]$ is rapidly decaying.

This is where the role of vanishing moments proves useful. It is seen from (4.5) that the zero-mean property of a wavelet is equivalent to the condition

$$\sum_{k \in \mathbf{Z}} g[k] = 0.$$

Using arguments similar to the ones used in Chapter 3, one can show that

$$\sum_{k \in \mathbf{Z}} |d * g[k]| < \infty.$$

This guarantees that the wavelet filters $g[k]$ and $g'[k]$ have comparable localizations.

(4) **Uniqueness**. Finally, note that although $\mathcal{H}\psi(x)$ is unique, the choice of the scaling function $\varphi'(x)$ and the wavelet filter $g'[k]$ that generate $\mathcal{H}\psi(x)$ is by no means unique. For example, if $\varphi'(x)$ and $g'[k]$ are such that (4.21) holds, then so does the choice

$$\varphi''(x) = \sum r[k]\varphi'(x-k) \quad \text{and} \quad g''[k] = g' * r_{\text{inv}}[k],$$

where $r[k]$ is such that its convolutional inverse $r_{\text{inv}}[k]$ is well-defined.

On the other, it was observed by Kingsbury that (approximate) Hilbert pairs of wavelets were obtained if one forced the lowpass filters of one channel to interpolate the lowpass filters of the second channel midway between its samples [24]. In the context of quadrature-mirror filter banks, this was later mathematically verified by Selesnick by considering wavelets as the limit functions defined by the infinite-product formula for the iterated filter bank [25].

4.5.3   Biorthogonal wavelet bases

The above construction can be directly extended to biorthogonal wavelet bases of $\mathbf{L}^2$, which includes orthonormal bases as a particular instance.

We recall from §4.2 that a biorthogonal wavelet basis is associated with a pair of distinct scaling functions $\varphi(x)$ and $\tilde{\varphi}(x)$. These are respectively the primal and dual scaling functions. Let primal and dual wavelets be given by (4.7) and (4.8). These together constitute the primary wavelets.

As before, we are required to identify a secondary biorthogonal basis corresponding to the primal-dual scaling function[3] $\varphi'(x)$ and $\tilde{\varphi}'(x)$ such that the corresponding primal-dual wavelets satisfy the criteria

$$\psi'(x) = \mathcal{H}\,\psi(x) \quad \text{and} \quad \tilde{\psi}'(x) = \mathcal{H}\,\tilde{\psi}(x). \tag{4.25}$$

Let the approximation order of $\varphi(x)$ and $\tilde{\varphi}(x)$ be $\alpha_1 + 1$ and $\alpha_2 + 1$. Therefore, for some τ_1 and τ_2, and distributions φ_1 and φ_2,

$$\varphi(x) = \beta_{\tau_1}^{\alpha_1} * \varphi_1(x) \qquad \text{and} \qquad \tilde{\varphi}(x) = \beta_{\tau_2}^{\alpha_2} * \varphi_2(x).$$

We consider the scaling functions

$$\varphi'(x) = \beta_{\tau_1 + 1/2}^{\alpha_1} * \varphi_1(x) \qquad \text{and} \qquad \tilde{\varphi}'(x) = \beta_{\tau_2 + 1/2}^{\alpha_2} * \varphi_2(x)$$

and the associated multiresolutions. Let the primal-dual wavelets be given by

$$\frac{1}{2}\psi'\left(\frac{x}{2}\right) = \sum g'[k]\varphi'(x - k)$$

and

$$\frac{1}{2}\tilde{\psi}'\left(\frac{x}{2}\right) = \sum \tilde{g}'[k]\tilde{\varphi}'(x - k).$$

The following result is then immediate from Theorem 18.

Corollary 19 (Hilbert pair of biorthogonal wavelets). *The following properties are equivalent.*

(i) The wavelets satisfy (5.5), and $(\tilde{\psi}'_{i,k})$ and $(\psi'_{i,k})$ constitute a biorthogonal wavelet basis.

(ii) The wavelet filters are related as $\tilde{g}'[k] = d * \tilde{g}[k]$ and $g'[k] = d * g[k]$.

The above construction also has the following properties:

(1) **System conditioning**. The biorthogonal systems have the same order, and hence the same Riesz bounds.

(2) **Biorthognality**. If the pair $\varphi(x)$ and $\tilde{\varphi}(x)$ form a biorthogonal pair, that is, if $\langle \tilde{\varphi}, \varphi(\cdot - k)\rangle = \delta[k]$, then so do the pair $\varphi'(x)$ and $\tilde{\varphi}'(x)$. To see this, note that

$$\mathcal{H}\,\beta_{\tau + 1/2}^{\alpha}(x) = -\Delta\beta_{\tau}^{\alpha}(x),$$

---

[3]We use ˜ to distinguish the primal and dual components of a given multiresolution, while we use ′ to distinguish the primary and secondary multiresolutions.

and that both $\mathscr{H}$ and Δ are unitary. Therefore, at least formally,

$$
\begin{aligned}
\langle \tilde{\varphi}', \varphi'(\cdot - k) \rangle &= \int (\beta^{\alpha_2}_{\tau_2 + 1/2} * \varphi_2)(x)\, (\beta^{\alpha_1}_{\tau_1 + 1/2} * \varphi_1)(x - k)\, dx \\
&= \int \mathscr{H}(\beta^{\alpha_2}_{\tau_2 + 1/2} * \varphi_2)(x)\, \mathscr{H}(\beta^{\alpha_1}_{\tau_1 + 1/2} * \varphi_1)(x - k)\, dx \\
&= \int \Delta(\beta^{\alpha_2}_{\tau_2} * \varphi_2)(x)\, \Delta(\beta^{\alpha_1}_{\tau_1} * \varphi_1)(x - k)\, dx \\
&= \langle \tilde{\varphi}, \varphi(\cdot - k) \rangle.
\end{aligned}
$$

This establishes the assertion.

(3) **Relation between filters**. The refinement filters on the primal and dual side are related as

$$
H'(z) = D(-z^{-1})H(z) \quad \text{and} \quad \tilde{H}(z^{-1}) = D(-z^{-1})\tilde{H}'(z^{-1}). \tag{4.26}
$$

where $D(z)$ is the z-transform of the filter in (4.13). On the other hand, the wavelet filters on the primal and dual side are related as

$$
G'(z) = D(z)G(z) \quad \text{and} \quad \tilde{G}(z^{-1}) = D(z)\tilde{G}'(z^{-1}). \tag{4.27}
$$

(4) **Perfect reconstruction**. If the analysis and synthesis filters of the original biorthogonal system satisfy the perfect reconstruction conditions,

$$
G(z^{-1})\tilde{G}(z) + H(z^{-1})\tilde{H}(z) = 1
$$

and

$$
G(z^{-1})\tilde{G}(-z) + H(z^{-1})\tilde{H}(-z) = 0,
$$

then so do the filters of the HT pair. Note that this is obvious when $z = 1$. On the other hand, for $z \neq 1$, we have from (4.26) and (4.27),

$$
\begin{aligned}
G'(z^{-1})\tilde{G}'(z) &+ H'(z^{-1})\tilde{H}'(z) \\
&= D(z^{-1})D(z)G(z^{-1})\tilde{G}'(z) + D(-z)D(-z^{-1})H(z^{-1})\tilde{H}(z) \\
&= G(z^{-1})\tilde{G}(z) + H(z^{-1})\tilde{H}(z).
\end{aligned}
$$

The first condition now follows from the identity $D(z)D(z^{-1}) = 1$ which holds when $z \neq 1$. The other condition can be similarly verified.

4.6   Hilbert pairs in higher dimensions

We now extend the one-dimensional constructing to two-dimensions. This is inspired by the construction of the dual-tree wavelets by Kingsbury et al. [27, 24, 26]. In particular, we use the conventional *separable* basis functions, that is,
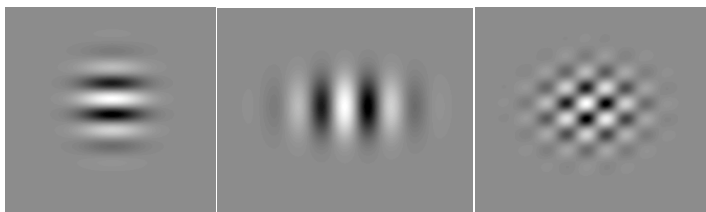
Figure 4.1: Wavelets associated with a separable basis. The figure shows the LH, HL and HH wavelets in the space domain.

functions realized using tensor products of one-dimensional scaling functions and wavelets. We use the directional Hilbert transform introduced in Chapter 2 as the two-dimensional extension of the Hilbert transform. It will be clear form the foregoing discussion that the construction can be extended to higher dimensions.

We recall that biorthogonal wavelet bases of $\mathbf{L}^2(\mathbf{R})$ can be combined to realize separable biorthogonal wavelet basis of $\mathbf{L}^2(\mathbf{R}^2)$ using straightforward tensor-products. The underlying principle is the following [14].

Theorem 20 (Separable biorthogonal bases). *Let $\psi_p(x)$ and $\tilde{\psi}_p(x)$ be the primal and dual wavelets of a biorthogonal wavelet basis of $\mathbf{L}^2(\mathbf{R})$, where the corresponding scaling functions are respectively given by $\varphi_p(x)$ and $\tilde{\varphi}_p(x)$. Similarly, let $\psi_q(x)$ and $\tilde{\psi}_q(x)$ be the wavelets of another biorthogonal basis with scaling functions $\varphi_q(x)$ and $\tilde{\varphi}_q(x)$. Consider the following system of wavelets*

$$\begin{aligned}
\psi_1(\boldsymbol{x}) &= \varphi_p(x_1)\psi_q(x_2), & \tilde{\psi}_1(\boldsymbol{x}) &= \tilde{\varphi}_p(x_1)\tilde{\psi}_q(x_2), \\
\psi_2(\boldsymbol{x}) &= \psi_p(x_1)\varphi_q(x_2), & \tilde{\psi}_2(\boldsymbol{x}) &= \tilde{\psi}_p(x_1)\tilde{\varphi}_q(x_2), \\
\psi_3(\boldsymbol{x}) &= \psi_p(x_1)\psi_q(x_2), & \tilde{\psi}_3(\boldsymbol{x}) &= \tilde{\psi}_p(x_1)\tilde{\psi}_q(x_2).
\end{aligned} \tag{4.28}$$

Then the dilations and translations of the wavelets on the left (primal wavelets) and the dilations and translations of the wavelets on the right side(dual wavelets) together constitute a biorthogonal wavelet basis of $\mathbf{L}^2(\mathbf{R}^2)$.

The functions $\psi_1(\boldsymbol{x}), \psi_2(\boldsymbol{x})$ and $\psi_3(\boldsymbol{x})$ are popularly referred to as the "low-high" (LH), "high-low" (HL) and "high-high" (HH) wavelets to emphasize the directions along which the "low pass" scaling functions and the "high pass" wavelets operate. In particular, note that if we denote the subspace

$$\mathrm{span}\Big\{f(x_1 - m)g(x_2 - n)\colon (m,n)\in \mathbf{Z}^2\Big\}$$

by $V(f)\otimes V(g)$, then the primal and dual approximation spaces for the above construction are respectively given by $V(\varphi_p)\otimes V(\varphi_q)$ and $V(\tilde{\varphi}_p)\otimes V(\tilde{\varphi}_q)$.

4.6.1 Motivation

Our construction of the two-dimensional wavelets is motivated by the concept of analytic functions and its connection with Hilbert transforms. A complex-valued function is said to be *analytic* if its Fourier transform (or spectrum) vanishes on one half of the real axis. For example, consider the function

$$\psi_a(x) = \psi(x) + j\,\mathcal{H}\,\psi(x). \tag{4.29}$$

Since

$$\hat{\psi}_a(\omega) = [1 + \operatorname{sign}(\omega)]\,\hat{\psi}(\omega),$$

it vanishes on the negative frequency axis. That the Hilbert transform appears in (4.29) is not a coincidence. It can be shown that (under appropriate hypothesis) the real and imaginary components of *every* function whose Fourier transform vanishes on the negative frequency axis form a Hilbert pair. This is the so-called *causality-analyticity* property.

In two-dimensions, we can have four equal partitions of the frequency plane called the *quadrants*, or two equal partitions called the *half-planes*. These are the natural generalizations of the concept of positive and negative frequencies in two-dimensions. We thus call a two-dimensional function analytic if its Fourier transform vanishes on the quadrants or the half-planes. Our main observation is that the real and imaginary components of such functions are related through the directional Hilbert transform. We recall that, corresponding to a unit vector u, the directional Hilbert transform $\mathcal{H}_u$ has the property that

$$\mathcal{H}_u f(\omega) = -j\operatorname{sign}(\omega^T u)\hat{f}(\omega). \tag{4.30}$$

In two-dimensions, we use $\mathcal{H}_\theta$ to denote the directional Hilbert transform corresponding to the unit vector where $u_\theta = (\cos\theta, \sin\theta)$. Note that if

$$\Psi_\theta(x) = \psi(x) + j\,\mathcal{H}_\theta\psi(x) \tag{4.31}$$

then

$$\hat{\Psi}_\theta(\omega) = \left[1 + \operatorname{sign}(\omega^T u_\theta)\right]\hat{\psi}(\omega).$$

Thus, the Fourier transform vanishes on the half space corresponding to frequencies ω for which $\omega^T u_\theta < 0$. It is not hard to see that every function which vanishes on some half-space or even a quadrant is of the form (4.31). This analytic property allows us to extend several key ideas associated with the one-dimensional construction to the higher dimensions.

We now show how such analytic wavelets can be designed in two-dimensions by combining conventional biorthogonal bases. The main idea is based on the construction given by Kingsbury et al. [26].

4.6.2   Analytic wavelets

We begin by setting up four separable biorthogonal wavelet bases using Theorem 20. The scaling functions of concern are $\varphi_p(x) = \varphi(x)$ and $\varphi_q(x) = \varphi'(x)$ that are associated with the analytic wavelet $\psi_a(x)$ in (4.29), where $\psi'(x) = \mathscr{H}\psi(x)$. This naturally leads to the possibility of four separable biorthogonal wavelet bases corresponding to the four approximation spaces, namely, $V(\varphi) \otimes V(\varphi)$, $V(\varphi) \otimes V(\varphi')$, $V(\varphi') \otimes V(\varphi)$ and $V(\varphi') \otimes V(\varphi')$. We will in fact use all of these to obtain a balanced construction.

The separable wavelets corresponding to the four scaling spaces are respectively

$$
\begin{aligned}
&\psi_1(x) = \varphi(x_1)\psi(x_2), &&\psi_4(x) = \varphi(x_1)\psi'(x_2),\\
&\psi_2(x) = \psi(x_1)\varphi(x_2), &&\psi_5(x) = \psi(x_1)\varphi'(x_2),\\
&\psi_3(x) = \psi(x_1)\psi(x_2), &&\psi_6(x) = \psi(x_1)\psi'(x_2),
\end{aligned}
$$

$$
\begin{aligned}
&\psi_7(x) = \varphi'(x_1)\psi(x_2), &&\psi_{10}(x) = \varphi'(x_1)\psi'(x_2),\\
&\psi_8(x) = \psi'(x_1)\varphi(x_2), &&\psi_{11}(x) = \psi'(x_1)\varphi'(x_2),\\
&\psi_9(x) = \psi'(x_1)\psi(x_2), &&\psi_{12}(x) = \psi'(x_1)\psi'(x_2). \qquad (4.32)
\end{aligned}
$$

This gives us the ensemble of primal wavelets $\psi_1(x), \cdots, \psi_{12}(x)$. The dual wavelets

$$
\tilde{\psi}_1(x), \cdots, \tilde{\psi}_{12}(x) \qquad (4.33)
$$

are specified identically, except using the dual wavelets instead of the primal ones. By judiciously using the one-sided spectrum of $\psi_a(x)$, and by combining the four separable wavelet bases in (4.32), we arrive at the following wavelet specifications

$$
\begin{aligned}
\Psi_1(x) &= \psi_a(x_1)\varphi(x_2) = \psi_2(x) + j\psi_8(x)\\
\Psi_2(x) &= \psi_a(x_1)\varphi'(x_2) = \psi_5(x) + j\psi_{11}(x)\\
\Psi_3(x) &= \varphi(x_1)\psi_a(x_2) = \psi_1(x) + j\psi_4(x)\\
\Psi_4(x) &= \varphi'(x_1)\psi_a(x_2) = \psi_7(x) + j\psi_{10}(x)\\
\Psi_5(x) &= \frac{1}{\sqrt{2}}\psi_a(x_1)\psi_a(x_2) = \left(\frac{\psi_3(x) - \psi_{12}(x)}{\sqrt{2}}\right) + j\left(\frac{\psi_6(x) + \psi_9(x)}{\sqrt{2}}\right)\\
\Psi_6(x) &= \frac{1}{\sqrt{2}}\psi_a^*(x_1)\psi_a(x_2) = \left(\frac{\psi_3(x) + \psi_{12}(x)}{\sqrt{2}}\right) + j\left(\frac{\psi_6(x) - \psi_9(x)}{\sqrt{2}}\right). \quad (4.34)
\end{aligned}
$$

The dual complex wavelets $\tilde{\Psi}_1(x), \cdots, \tilde{\Psi}_6(x)$ are specified in an identical fashion using the wavelets in (4.33). Note that the construction is complete in the sense that it involves all the $4 \times 3 = 12$ separable wavelets of the four multiresolutions. The factor $1/\sqrt{2}$ in (4.34) is used to ensure normalization—all the six complex wavelets have the same $\mathbf{L}^2(\mathbf{R}^2)$ norm. The key feature of the construction is the following.

Proposition 21 (Hilbert pairs). *The real and imaginary components of the complex wavelets in (4.34) form directional Hilbert transform pairs. In particular,*

$$\mathbf{Im}[\Psi_l(x)] = \mathcal{H}_{\theta_l}\mathbf{Re}[\Psi_l(x)] \tag{4.35}$$

where

$$\theta_1 = \theta_2 = 0, \quad \theta_3 = \theta_4 = \pi/2, \quad \theta_5 = \pi/4, \ \text{and} \ \theta_6 = 3\pi/4.$$

In other words, the components of the complex wavelets in (4.34) are related by directional Hilbert transforms along the four primal directions, namely, the horizontal, the vertical, and the two diagonals.

Proof. Note that it is sufficient to establish the correspondences for the wavelets $\Psi_1(x)$ and $\Psi_5(x)$; the rest follows from symmetry. The correspondence for the former is direct,

$$\mathcal{H}_0\mathbf{Re}[\Psi_1(x)] = \mathcal{H}[\psi(x_1)]\varphi(x_2) = \psi'(x_1)\varphi(x_2) = \mathbf{Im}[\Psi_1(x)].$$

We now show that

$$\mathbf{Im}[\Psi_5(x)] = \mathcal{H}_{\pi/4}\mathbf{Re}[\Psi_5(x)]. \tag{4.36}$$

Note that

$$\widehat{\mathbf{Re}[\Psi_5]}(\omega) = \frac{1}{\sqrt{2}}\left(\hat{\psi}_3(\omega) - \hat{\psi}_{12}(\omega)\right)$$

$$= \frac{1}{\sqrt{2}}\left(1 + \mathrm{sign}(\omega_1)\,\mathrm{sign}(\omega_2)\right)\hat{\psi}(\omega_1)\hat{\psi}(\omega_2),$$

and that

$$\widehat{\mathbf{Im}[\Psi_5]}(\omega) = \frac{1}{\sqrt{2}}\left(\hat{\psi}_6(\omega) + \hat{\psi}_9(\omega)\right)$$

$$= -\frac{j}{\sqrt{2}}\left(\mathrm{sign}(\omega_1) + \mathrm{sign}(\omega_2)\right)\hat{\psi}(\omega_1)\hat{\psi}(\omega_2).$$

Therefore, by (4.30), the Fourier transform of the right side of (4.36) is

$$-j\,\mathrm{sign}(\omega_1 + \omega_2)\,\frac{1}{\sqrt{2}}\left(1 + \mathrm{sign}(\omega_1)\,\mathrm{sign}(\omega_2)\right)\hat{\psi}(\omega_1)\hat{\psi}(\omega_2).$$

But this is exactly the Fourier transform of the left side of (4.36), since

$$\mathrm{sign}(\omega_1) + \mathrm{sign}(\omega_2) = \mathrm{sign}(\omega_1 + \omega_2)\left(1 + \mathrm{sign}(\omega_1)\,\mathrm{sign}(\omega_2)\right)$$

This establishes (4.36), and hence the theorem.   □

4.6.3 Orientation selectivity

Separable wavelets have limited orientation selectivity. As shown in Figure 4.1, the wavelets are oriented along the horizontal, vertical, and the two diagonals. The main difficulty is that the HH wavelet is not oriented purely along one direction. It has a checkerboard appearance with simultaneous pulsation along the diagonal directions. To decouple these directions, we use the causality-analyticity property.

Note that the difficulty is due to the fact that the HH wavelet is realized as the tensor product $\psi(x_1)\psi(x_2)$. As a result, its spectrum is not localized to any fixed quadrant or half-plane. On the other hand, note that if analytic wavelets of the form $\psi_a(x_1)\psi_a(x_2)$ is used, then it spectrum will be supported on a quadrant. As a result, the real and imaginary components of these wavelets have better orientation selectivity.

Chapter 5

Gabor-like wavelet transforms

Abstract — In this chapter, we use the framework developed in the previous chapter to identify two spline multiresolutions where the wavelet bases are related through the Hilbert transform. In effect, this gives us a family of analytic spline wavelets which are continuously indexed by their order. The remarkable property is that these analytic spline wavelets converge to a Gaussian-windowed complex exponential (Gabor function) as their order gets large. This gives us a finite-order approximation of the Gabor function, which is known to be optimally localized in the joint space-frequency domain.

In higher dimensions, we construct a family of complex spline wavelets that resemble the directional Gabor functions proposed by Daugman [3].

We also present a fast filterbank implementation of the associated complex wavelet transform using the fast Fourier transform[1].

5.1   Introduction

The need for combined space-frequency representations stemmed from the inadequacy of either space or frequency domain analysis to fully describe the nature of non-stationary signals. This led Gabor and others to consider the space-frequency representation of signals as a tool for measuring the frequency content of the signal at different points in space [2, 53]. One of the first steps in this direction was Gabor's reformulation of Heisenberg's Uncertainty Principle in terms of the so-called quantum law for information, the principle that *"the joint space-frequency domain of signals is quantized and the support of signals in this domain always exceed a certain*

---

[1]Parts of this chapter is based on the article [49]: K. N. Chaudhury, M. Unser, "Construction of Hilbert transform pairs of wavelet bases and Gabor-like transforms," IEEE Transactions on Signal Processing, vol. 57, no. 9, pp. 3411-3425, September 2009.

minimum". He went on to discover that it is the Gaussian-windowed complex exponentials (and their translates) which attain this minimum.

This discovery led him to introduce the *Gabor transform* in his famous 1946 paper [2]. The idea was to represent a non-stationary signal using the space and frequency translates of a Gaussian window. These *Gabor atoms* were of the form

$$G_{m,n}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-mX)^2}{2\sigma^2}\right] \exp\left[jn\Omega(x-mX)\right]$$

where the parameters X and Ω controlled the space and frequency resolution, respectively. The ultimate goal was to have the representation

$$f(x) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} c_{m,n} G_{m,n}(x).$$

While Gabor did give an iterative method for estimating the coefficients $c_{m,n}$, an analytic method for computing them was not known until Baastians provided a solution in 1980 [4]. The main difficult in computing the coefficients stems from the fact that the system $(G_{m,n})$ is not orthogonal in $\mathbf{L}^2$. Unfortunately, as predicted by the Balian-Low theorem, it does not even form a stable Riesz basis of $\mathbf{L}^2$ [54]. One can, of course, have Gabor frames. In this case, the reconstruction process, however, involves the computation of the dual frame, and this turns out to be computationally expensive and often unstable [4, 5].

5.2   Gabor-like wavelets

As an application of the ideas developed in Chapter 4, we now construct two wavelet bases $(\psi_{i,k})$ and $(\psi'_{i,k})$ within the framework of multiresolution analysis, where

$$\psi'_{i,k}(x) = \mathcal{H}\psi_{i,k}(x).$$

These wavelets are indexed by their order (or degree). The key feature of the analytic wavelet

$$\Psi(x) = \psi(x) + j\psi'(x)$$

is that it asymptotically converges to a Gabor function as the order gets large. This allows us to realize a fast *Gabor-like transform* using the standard filterbank algorithms. The difference is that, instead of the uniform space-frequency tiling used in Gabor analysis, we use the non-uniform space-frequency tiling used in wavelet theory. As is well-known, the latter "variable-scale" analysis has the advantage that it allows us to locate high-frequency components more accurately in space.

5.2.1 The B-spline multiresolution

We recall from Chapter 4 that the fractional B-spline $\beta_\tau^\alpha(x)$ is specified by the Fourier transform

$$\hat{\beta}_\tau^\alpha(\omega) = \left(\frac{1 - e^{-j\omega}}{j\omega}\right)^p \left(\frac{1 - e^{j\omega}}{-j\omega}\right)^q \tag{5.1}$$

where

$$p = \frac{1}{2}(\alpha + 1) + \tau \quad \text{and} \quad q = \frac{1}{2}(\alpha + 1) - \tau.$$

It is a valid scaling function for all $\alpha \geq 0$ and τ. It satisfies the two-scale relation

$$\frac{1}{2}\beta_\tau^\alpha\left(\frac{x}{2}\right) = \sum_{k \in \mathbf{Z}} b_\tau^\alpha[k]\beta_\tau^\alpha(x - k)$$

where

$$H_\tau^\alpha(e^{j\omega}) = \left(\frac{1 + e^{-j\omega}}{2}\right)^p \left(\frac{1 + e^{j\omega}}{2}\right)^q.$$

Consider the multiresolution generated by $\beta_\tau^\alpha(x)$, namely, the sequence of subspaces

$$\{0\} \subset \cdots \subset \mathbf{V}_1 \subset \mathbf{V}_0 \subset \mathbf{V}_{-1} \subset \cdots \subset \mathbf{L}^2(\mathbf{R})$$

where

$$\mathbf{V}_i = \mathrm{span}\left\{\beta_\tau^\alpha(2^{-i}x - k) : k \in \mathbf{Z}\right\}.$$

The important observation is that the translates of $\beta_\tau^\alpha(x)$ are not orthogonal, excepting when $\alpha = 0$ and $\tau = 1/2$ (Haar system). However, the translates of $\beta_\tau^\alpha(x)$ do form a Riesz basis of $\mathbf{V}_0$. To see this, consider the autocorrelation sequence

$$a_\tau^\alpha[k] = \int \beta_\tau^\alpha(x)\beta_\tau^\alpha(x - k)\,dx \qquad (k \in \mathbf{Z})$$

and its Fourier transform

$$A_\tau^\alpha(e^{j\omega}) = \sum_{k \in \mathbf{Z}} a_\tau^\alpha[k]e^{-j\omega k} = \sum_{n \in \mathbf{Z}} |\hat{\beta}_\tau^\alpha(\omega + 2\pi n)|^2.$$

It can be verified that $B < A_\tau^\alpha(e^{j\omega}) < C$ for constants $B, C > 0$, e.g., see [22]. This is sufficient for the translates to form a Riesz basis of their span [14].

Within this multiresolution, the *B-spline wavelet* is given by

$$\frac{1}{2}\psi_\tau^\alpha\left(\frac{x}{2}\right) = \sum_{k \in \mathbf{Z}} g_\tau^\alpha[k]\beta_\tau^\alpha(x - k) \tag{5.2}$$

where

$$G_\tau^\alpha(e^{j\omega}) = e^{j\omega} A(-e^{j\omega}) H_\tau^\alpha(-e^{-j\omega}).$$

The wavelet filter is so chosen that $\psi_\tau^\alpha(x)$ is orthogonal to the translates of $\beta_\tau^\alpha(x)$ (and its dilates), but not to its own translates [55]. Thus, the translates of the wavelet form a Riesz basis of $\mathbf{W}_0$, which, in this case, is the orthogonal complement of $\mathbf{V}_0$ in $\mathbf{V}_{-1}$.

The dual multiresolution is specified using the dual $\tilde{\beta}_\tau^\alpha(x)$. This is specified by the biorthogonality requirement

$$\int \tilde{\beta}_\tau^\alpha(x) \beta_\tau^\alpha(x - n) = \delta[n] \qquad (n \in \mathbf{Z}).$$

Clearly, $\tilde{\beta}_\tau^\alpha(x)$ is non-unique. However, by requiring it to be in $\mathbf{V}_0$, we recover an unique dual given by

$$\hat{\tilde{\beta}}_\tau^\alpha(\omega) = A_\tau^\alpha(e^{j\omega})^{-1} \hat{\beta}_\tau^\alpha(\omega). \tag{5.3}$$

Note that the expression on the right is well-defined since $A_\tau^\alpha(e^{j\omega})$ is non-vanishing.

Notice that the choice of the dual scaling function fixes the dual refinement filter. The wavelet filter for the dual multiresolution is then automatically fixed by the perfect-reconstruction conditions [56]. In particular, the complete system of filters is given by

$$\left.\begin{aligned}
H_\tau^\alpha(e^{j\omega}) &= \left(\frac{1 + e^{-j\omega}}{2}\right)^{\frac{\alpha+1}{2}+\tau} \left(\frac{1 + e^{j\omega}}{2}\right)^{\frac{\alpha+1}{2}-\tau} \\
G_\tau^\alpha(e^{j\omega}) &= e^{j\omega} A_\tau^\alpha(-e^{j\omega}) H_\tau^\alpha(-e^{-j\omega}) \\
\tilde{H}_\tau^\alpha(e^{j\omega}) &= \frac{H_\tau^\alpha(e^{j\omega}) A_\tau^\alpha(e^{j\omega})}{A_\tau^\alpha(e^{j2\omega})} \\
\tilde{G}_\tau^\alpha(e^{j\omega}) &= \frac{G_\tau^\alpha(e^{j\omega})}{A_\tau^\alpha(e^{j2\omega}) A_\tau^\alpha(-e^{j\omega})}
\end{aligned}\right\} \tag{5.4}$$

This gives us the dual wavelet

$$\frac{1}{2} \tilde{\psi}_\tau^\alpha\left(\frac{x}{2}\right) = \sum_{k \in \mathbf{Z}} \tilde{g}_\tau^\alpha[k] \tilde{\beta}_\tau^\alpha(x - k).$$

5.2.2   Hilbert pairs of spline wavelets

For every fixed α and τ, we have the above multiresolution structure involving the B-spline wavelet and its dual. From Corollary 4 of Chapter 4, we deduce the following.

Theorem 22 (Hilbert pairs)**.** *The Hilbert transform of a B-spline wavelet is again a B-spline wavelet of same order, but with a different shift. In particular,*

$$\mathcal{H}\,\psi_\tau^\alpha(x) = \psi_{\tau+1/2}^\alpha(x). \tag{5.5}$$

Similarly, for the dual wavelet,

$$\mathcal{H}\,\tilde{\psi}_\tau^\alpha(x) = \tilde{\psi}_{\tau+1/2}^\alpha(x). \tag{5.6}$$

Proof. Following the notations of Corollary 4 (Chapter 4), we have

$$\varphi(x) = (\beta_\tau^\alpha * \varphi_1)(x) \quad \text{and} \quad \varphi'(x) = (\beta_{\tau+1/2}^\alpha * \varphi_1)(x)$$

where $\varphi_1(x)$ is simply the Dirac distribution. Similarly,

$$\tilde{\varphi}(x) = (\beta_\tau^\alpha * \varphi_2)(x) \quad \text{and} \quad \tilde{\varphi}'(x) = (\beta_{\tau+1/2}^\alpha * \varphi_2)(x)$$

where

$$\varphi_2(x) = \sum q[k]\delta(x-k) \quad \text{and} \quad \sum q[k]e^{-j\omega k} = A_\tau^\alpha(e^{j\omega})^{-1}.$$

Here we have used the fact that $A_\tau^\alpha(e^{j\omega})$, and hence $q[k]$, does not depend on τ. By Corollary 4, we know that (5.5) and (5.6) holds if

$$g'[k] = d * g[k] \quad \text{and} \quad \tilde{g}'[k] = d * \tilde{g}[k].$$

It is straightforward to verify from (5.4) that this is indeed the case.   □

The importance of this result is that it gives us the family of analytic B-spline wavelets

$$\Psi_\tau^\alpha(x) = \psi_\tau^\alpha(x) + j\psi_{\tau+1/2}^\alpha(x) \qquad (\alpha \geq 0, \tau \in \mathbf{R}). \tag{5.7}$$

Note that α controls the smoothness and the number of vanishing moments (oscillations) of $\psi_\tau^\alpha(x)$ [55]. In particular, when α is sufficiently large, $\psi_\tau^\alpha(x)$ satisfies the hypothesis of Theorem 2 in Chapter 2. In this regime, the prediction is that its Hilbert transform has comparable localization, oscillation and smoothness. The following asymptotic analysis confirms this symmetry.

5.2.3   Asymptotic behavior

The asymptotic nature of $\psi_\tau^\alpha(x)$ can be readily guessed from that of the symmetric polynomial B-spline wavelets for which α is an integer and $\tau = 0$. It was shown in [20] that these wavelets asymptotically behave as modulated Gaussians as α gets large:

$$\psi_0^\alpha(x) \sim \varphi(x)\cos(\omega_0 x + \phi), \tag{5.8}$$

where $\varphi(x)$ is Gaussian, and ω_0 and ϕ are appropriate modulation parameters. Now, it can be shown that

$$\psi_\tau^\alpha(x) = \mathcal{H}_{-\tau}\psi_0^\alpha(x)$$

where $\mathcal{H}_{-\tau}$ is the fractional Hilbert transform of shift $-\tau$ (we will derive this in Chapter 6 in a different context). Form the modulation property of $\mathcal{H}_{-\tau}$ (cf. Chapter 2), it follows that

$$\lim_{\alpha \longrightarrow \infty} \psi_\tau^\alpha(x) \approx \varphi(x)\cos\left(\omega_0 x + \phi - \pi\tau\right).$$

For the analytic wavelets in (5.7), we conclude that

$$\lim_{\alpha \longrightarrow \infty} \Psi_\tau^\alpha(x) \approx \varphi(x)\cos\left(\omega_0 x + \phi - \pi\tau\right) + \varphi(x)\cos\left(\omega_0 x + \phi - \pi(\tau + 1/2)\right)$$
$$= \varphi(x)\exp\left(\omega_0 x + \phi - \pi\tau\right).$$

Note that we use the approximation symbol because the modulation property holds only approximately for Gaussian windows. The following result shows that this approximation is quite accurate.

Theorem 23 (Gabor-like wavelets). *The analytic B-spline wavelet $\Psi_\tau^\alpha(x)$ converges to a Gabor function as α get large. In particular,*

$$\Psi_\tau^\alpha(x) \sim \frac{2M^{\alpha+1}\sigma}{\sqrt{2\pi(\alpha+1)}}\exp\left[-\frac{\sigma^2(x-1/2)^2}{2(\alpha+1)}\right]\exp\left[j(\omega_0 x - \omega_0/2 - \pi\tau)\right] \quad (\alpha \longrightarrow \infty)$$

(5.9)

where $M = 0.670$, $\omega_0 = 5.142$, and $\sigma = 2.670$.

We provide a proof in Appendix A. This is adapted from the proof in [20].

The above convergence in fact happens quite rapidly. For instance, we have observed that the joint time-frequency resolution of the complex cubic B-spline wavelet ($\alpha = 3$) is already within 3% of the limit specified by the uncertainty principle. Figure 5.1 depicts the complex wavelets generated using Hilbert pairs of B-spline wavelets. Also shown in the figure is the envelope of the complex wavelet which closely resembles the well-localized window of the Gabor function.

Note that while the B-spline wavelets tend to be optimally localized in space, they are not orthogonal to their translates. The reconstruction therefore requires the use of dual (synthesis) functions. The flip side is that these dual-spline wavelets have a comparatively poor spatial localization, that deteriorates with the increase in degree. This is evident from Figure 5.2, which shows quadrature pairs of such wavelets of different degrees. However, the dual wavelets have the same rate of decay as their analysis counterpart, and the reconstruction algorithm is fast and numerically stable.

(a) $\alpha = 3$

(b) $\alpha = 6$
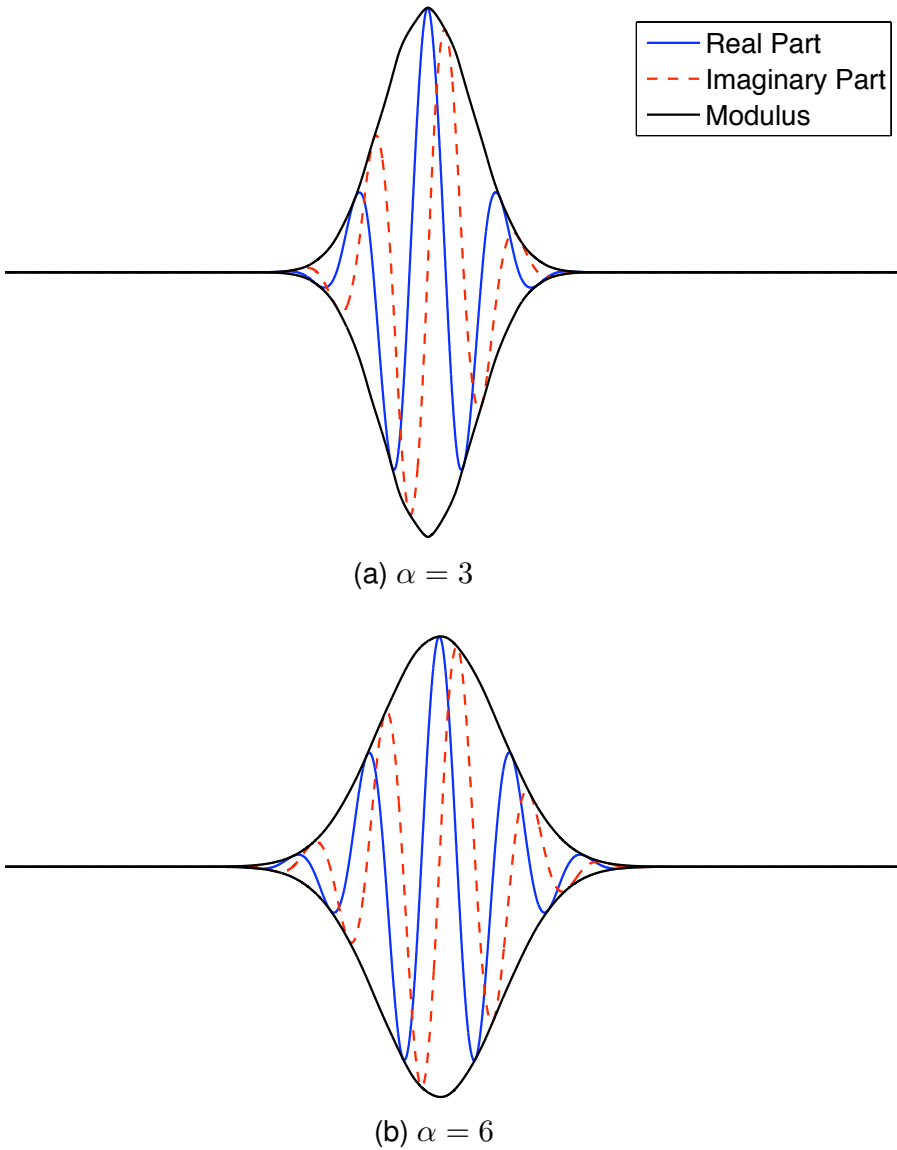
Figure 5.1: Hilbert pairs of B-spline wavelets. In either case, Blue (solid line): $\psi_0^\alpha(x)$, Red (broken line): $\psi_{1/2}^\alpha(x)$, Black (solid line): $|\psi_0^\alpha(x) + \psi_{1/2}^\alpha(x)|$
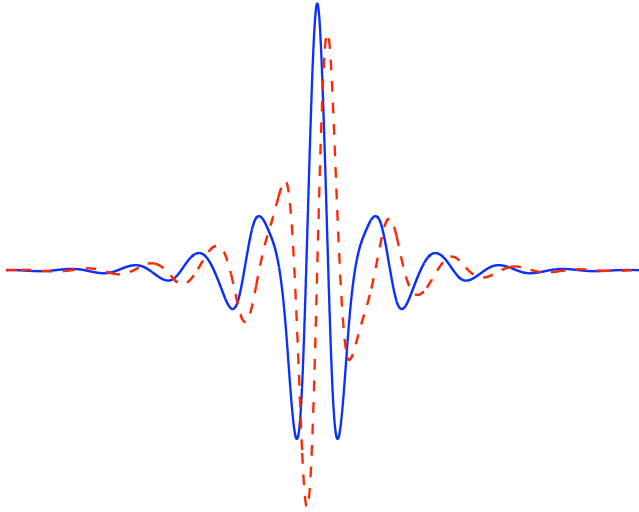
(a) $\alpha = 3$
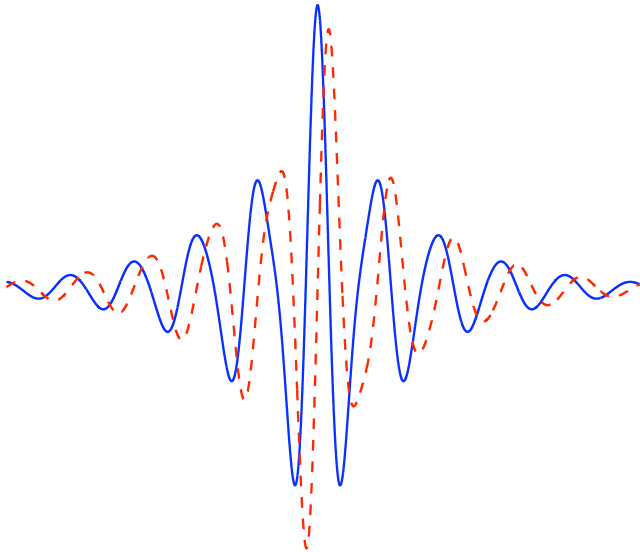
(b) $\alpha = 6$

Figure 5.2: Hilbert pairs of dual-spline wavelets. Blue (solid line): $\psi_0^\alpha(x)$, Red (broken line): $\psi_{1/2}^\alpha(x)$.

5.3 Gabor-like transform

We now discuss the implementation of the discrete wavelet transforms corresponding to the Hilbert pairs in (5.5). We take the order to be sufficiently large α, say, $\alpha = 6$, while we arbitrarily set τ. Our eventual goal is to combine the coefficients of the two transforms to realize the coefficients

$$c_i[k] = \int f(x)\Psi_{i,k}(x)\,dx \qquad (i,k \in \mathbf{Z}).$$

In view of (5.9), we call this the *Gabor-like wavelet transform*. The main difference with (??) is that instead of the uniform space-frequency tiling provided by the Gabor transform, we use the logarithmic space-frequency tiling provided by the wavelet transform.

Note that this requires two biorthogonal multiresolutions generated by the B-spline and its dual. In particular, the primary multiresolutions are generated by

$$\varphi(x) = \beta_\tau^\alpha(x) \quad \text{and} \quad \tilde{\varphi}(x) = \tilde{\beta}_\tau^\alpha(x).$$

The secondary multiresolutions are generated by

$$\varphi(x) = \beta_{\tau+1/2}^\alpha(x) \quad \text{and} \quad \tilde{\varphi}(x) = \tilde{\beta}_{\tau+1/2}^\alpha(x).$$

The perfect-reconstruction filters associated with these biorthogonal systems are given by (5.4). While the degree α is same in either case, we set the shifts as τ and $\tau + 1/2$ for the primary and secondary systems, respectively.

5.3.1 Pre-filter

To achieve a coherent analysis, we need to project the input signal onto the approximation spaces $\mathbf{V}_0(\varphi)$ and $\mathbf{V}_0(\varphi')$. In particular, given a finite-energy signal $f(x)$, we consider its orthogonal projection onto $\mathbf{V}_0(\varphi)$ given by

$$f_0(x) = \sum_{k \in \mathbf{Z}} c_0[k]\varphi(x-k)$$

where

$$c_0[k] = \int f(x)\tilde{\varphi}(x-k)\,dx. \tag{5.10}$$

We take $\hat{\varphi}(x)$ to be the unique dual in $\mathbf{V}_0(\varphi)$ specified by (5.3).

The wavelet decomposition of $f_0(x)$ at resolutions $j = 1, \cdots, J$ is then given by

$$f_0(x) = \sum_{k \in \mathbf{Z}} c_J[k]\varphi_{J,k}(x) + \sum_{1 \leq j \leq J}\sum_{k \in \mathbf{Z}} d_j[k]\psi_{j,k}(x) \tag{5.11}$$

where $d_j[k]$ and $c_J[k]$ are obtained from $c_0[k]$ using the standard filterbank algorithm of Mallat [51]. Here $d_j[k]$ are the *wavelet coefficients*, and $c_J[k]$ are the *approximation coefficients* corresponding to the coarsest scale.

In practice, one only has access to the discrete samples of the signal. Let $f[k]$ be such (uniform) samples. It turns out that by assuming the signal to be bandlimited, a particularly simple filtering algorithm for computing the projection coefficients is obtained. This is given by

$$c_0[k] = (f * p)[k] \tag{5.12}$$

where

$$C_0(e^{j\omega}) = \hat{\varphi}(\omega)^* \qquad (-\pi < \omega \leq \pi).$$

That is,

$$P(e^{j\omega}) = \sum_{n \in \mathbf{Z}} \text{rect}\left(\frac{\omega + 2\pi n}{2\pi}\right) \hat{\varphi}(\omega + 2\pi n)^*. \tag{5.13}$$

To see this, note that, applying Poisson's summation formula to (5.10), we have

$$C_0(e^{j\omega}) = \sum_{n \in \mathbf{Z}} \hat{f}(\omega + 2\pi n)\hat{\varphi}(\omega + 2\pi n)^*.$$

Let $F(e^{j\omega})$ be the Fourier transform of $f[k]$. Since $f(x)$ is bandlimited,

$$\hat{f}(\omega) = F(e^{j\omega})\text{rect}\left(\frac{\omega}{2\pi}\right).$$

Therefore, by (5.13),

$$C_0(e^{j\omega}) = F(e^{j\omega}) \sum_{n \in \mathbf{Z}} \text{rect}\left(\frac{\omega + 2\pi n}{2\pi}\right) \hat{\varphi}(\omega + 2\pi n)^* = F(e^{j\omega})P(e^{j\omega}).$$

This is exactly (5.12). For the secondary multiresolution, we use the pre-filter

$$c_0'[k] = (f * p')[k] \tag{5.14}$$

where

$$P'(e^{j\omega}) = \hat{\varphi}'(\omega)^* \qquad (-\pi < \omega \leq \pi).$$

Note that both $P(e^{j\omega})$ and $P'(e^{j\omega})$ are non-vanishing on $(-\pi, \pi]$. This means that the projection operations are perfectly reversible.

5.3.2   Analysis

Henceforth, we confine our analysis to discrete periodic signals. We denote such a signal of size N by $\mathbf{f} = (f_1, \ldots, f_N)$. The orthogonal projections are then given by

$$c_0^L = \mathbf{P}\mathbf{f} \quad \text{and} \quad c_0'^L = \mathbf{P}'\mathbf{f}$$

where $\mathbf{P}$ and $\mathbf{P}'$ are the circulant matrices that correspond to the discrete versions of the pre-filters in (5.12) and (5.14). In practice, these are only specified implicitly, since we do the pre-filtering in the frequency domain. The reason for using the notation c_0^L instead of c_0 will be clear shortly.

Let $\mathbf{U}_h$ denote the composition of the circulant matrix corresponding to the filter $h[k]$ and the downsampling matrix. For the primary system, the low pass and the high pass subbands at successive scales $i = 1, \ldots, J$ are given by the recursion

$$c_i^L = \mathbf{U}_{\tilde{h}} c_{i-1}^L \qquad c_i^H = \mathbf{U}_{\tilde{g}} c_{i-1}^L \qquad (i = 1, 2, \ldots, J).$$

Similarly, for the secondary system,

$$c_i'^L = \mathbf{U}_{\tilde{h}} c_{i-1}'^L \qquad c_i'^H = \mathbf{U}_{\tilde{g}} c_{i-1}'^L.$$

The complex wavelet subbands are then given by

$$\mathbf{w}_i = c_i^H + j c_i'^H \qquad (1 \leq i \leq J).$$

We remark that the projection filters play a crucial role as far as *analyticity* is concerned. The Fourier transform of the analytic wavelet is one-sided by construction. The corresponding complex wavelet filter,

$$\tilde{G}(e^{j\omega}) + j \tilde{G}'(e^{j\omega}),$$

however does not inherit this property naturally. It is only for the combination of the projection and wavelet filters,

$$G_a(e^{j\omega}) = P(e^{j\omega}) \tilde{G}(e^{j\omega}) + j P'(e^{j\omega}) \tilde{G}'(e^{j\omega}),$$

that $G_a(e^{j\omega}) = 0$ when $-\pi < \omega \leq 0$. Figure 5.3 shows the one-sided response of the filter.

5.3.3   Reconstruction

Clearly, the transformation

$$\mathbf{T} : \mathbf{f} \mapsto (c_J^L, c_J'^L, \mathbf{w}_1, \ldots, \mathbf{w}_J) \qquad (5.15)$$

is redundant by a factor of two. However, note that $\mathbf{T}$ is realized by concatenating two bases, and is hence one-to-one, that is, $\mathbf{Tf}_1 = \mathbf{Tf}_2$ if and only if $\mathbf{f}_1 = \mathbf{f}_2$. This means that $\mathbf{T}$ has non-unique left-inverses $\mathbf{T}^\dagger$ for which

$$\mathbf{f} = \mathbf{T}^\dagger(\mathbf{Tf}).$$

Let $\mathbf{V}_h$ denote the composition of the upsampling matrix and the circulant matrix corresponding to the filter $h[k]$. In our case, we use the simple left-inverse

$$\mathbf{T}^\dagger : (c_J^L, c_J'^L, \mathbf{w}_1, \ldots, \mathbf{w}_J) \mapsto \mathbf{f} = \frac{1}{2}[\mathbf{P}^{-1} c_0^L + \mathbf{P}'^{-1} c_0'^L].$$
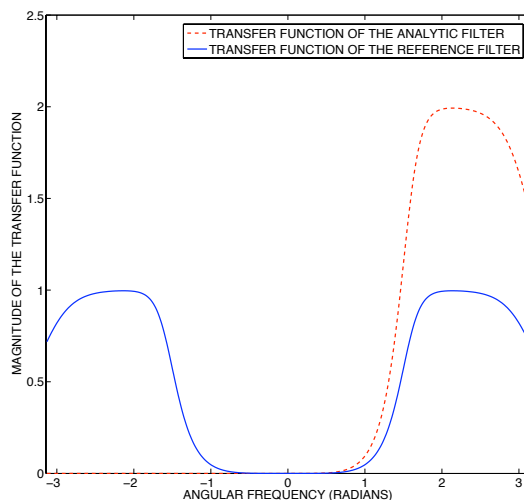
Figure 5.3: Transfer function of the *analytic* wavelet filter $G_a(e^{j\omega})$.

where c_0^L and c'^L_0 are obtained through the recursion

$$c_i^L = \mathbf{U}_h c_{i+1}^L + \mathbf{V}_g \mathbf{Re}(\mathbf{w}_{i+1})$$
$$c'^L_i = \mathbf{U}_{h'} c'^L_{i+1} + \mathbf{V}_{g'} \mathbf{Im}(\mathbf{w}_{i+1}) \qquad (i = J - 1, \ldots, 0).$$

In short, the reconstruction amounts to inverting the two wavelet transforms and averaging the outputs.

We implement the transform for finite periodic data using the fast Fourier transform. The basic algorithm is the same as the one used in [57]. This is briefly explained in Appendix C.

5.4 Directional Gabor-like wavelets

We are interested in two-dimensional Gabor functions of the form

$$G(\mathbf{x}) = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left[-\frac{(x_1 - u)^2}{2\sigma_1^2} - \frac{(x_2 - v)^2}{2\sigma_2^2}\right] \exp\left[j(\xi_1 x_1 + \xi_2 x_2)\right]. \quad (5.16)$$

This are simply complex plane waves localized using Gaussian window. The direction of the plane wave (or oscillation) is determined by ξ_1 and ξ_2. The cases of interest are the following: $\xi_2 = 0$ (wave along x_1), $\xi_1 = 0$ (wave along x_2), $\xi_1 = \pm\xi_2$ (wave along one of the diagonals).

We now approximate such direction-selective Gabor functions using the spline multiresolution introduced in §5.2. To do so, we use the two-dimensional complex wavelets of Chapter 4. In view of the correspondence in (5.5), we fix some α and τ, and set

$$\varphi(x) = \beta_\tau^\alpha(x) \quad \text{and} \quad \varphi'(x) = \beta_{\tau+1/2}^\alpha(x).$$

The corresponding wavelets are

$$\psi(x) = \psi_\tau^\alpha(x) \quad \text{and} \quad \psi'(x) = \psi_{\tau+1/2}^\alpha(x).$$

This fixes the six oriented complex wavelets $\Psi_1(x; \alpha, \tau), \ldots, \Psi_6(x; \alpha, \tau)$. We recall that the real and imaginary components of these wavelets form Hilbert pairs,

$$\mathbf{Im}[\Psi_l(x; \alpha, \tau)] = \mathcal{H}_{\theta_l}\mathbf{Re}[\Psi_l(x; \alpha, \tau)] \tag{5.17}$$

where

$$\theta_1 = \theta_2 = 0, \quad \theta_3 = \theta_4 = \pi/2, \quad \theta_5 = \pi/4, \text{ and } \theta_6 = 3\pi/4.$$

As argued in Chapter 4, the above correspondences allude to the fact that the wavelets are selectively oriented along the directions $\theta_1, \ldots, \theta_6$. This becomes apparent from the following asymptotic characterization.

Proposition 24 (Directional Gabor-like wavelets). *The wavelets $\Psi_k(x; \alpha, \tau)$ converge to Gabor functions as α gets large,*

$$\lim_{\alpha \longrightarrow \infty} \Psi_1(x; \alpha, \tau) \sim M_1 \exp\left[-\frac{(x-1/2)^2}{\sigma_1^2} - \frac{(y-\tau)^2}{\sigma_2^2} \right] \exp\left[j(\omega_0 x - \pi\tau) \right]$$

$$\lim_{\alpha \longrightarrow \infty} \Psi_2(x; \alpha, \tau) \sim M_1 \exp\left[-\frac{(x-1/2)^2}{\sigma_1^2} - \frac{(y-\tau-1/2)^2}{\sigma_2^2} \right] \exp\left[j(\omega_0 x - \pi\tau) \right]$$

$$\lim_{\alpha \longrightarrow \infty} \Psi_3(x; \alpha, \tau) \sim M_1 \exp\left[-\frac{(x-\tau)^2}{\sigma_2^2} - \frac{(y-1/2)^2}{\sigma_1^2} \right] \exp\left[j(\omega_0 y - \pi\tau) \right]$$

$$\lim_{\alpha \longrightarrow \infty} \Psi_4(x; \alpha, \tau) \sim M_1 \exp\left[-\frac{(x-\tau-1/2)^2}{\sigma_2^2} - \frac{(y-1/2)^2}{\sigma_1^2} \right] \exp\left[j(\omega_0 y - \pi\tau) \right]$$

$$\lim_{\alpha \longrightarrow \infty} \Psi_5(x; \alpha, \tau) \sim M_2 \exp\left[-\frac{(x-1/2)^2}{\sigma_1^2} - \frac{(y-1/2)^2}{\sigma_1^2} \right] \exp\left[j(\omega_0(x+y) - 2\pi\tau) \right]$$

$$\lim_{\alpha \longrightarrow \infty} \Psi_6(x; \alpha, \tau) \sim M_2 \exp\left[-\frac{(x-1/2)^2}{\sigma_1^2} - \frac{(y-1/2)^2}{\sigma_1^2} \right] \exp\left[j\omega_0(y-x) \right],$$

$$\tag{5.18}$$

where $M_1, M_2, \sigma_1, \sigma_2,$ and ω_0 are appropriate parameters.

In Appendix B, we shown that

$$\lim_{\alpha \longrightarrow \infty} \beta_\tau^\alpha(x) \sim \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\tau)^2}{2\sigma^2} \right] \tag{5.19}$$
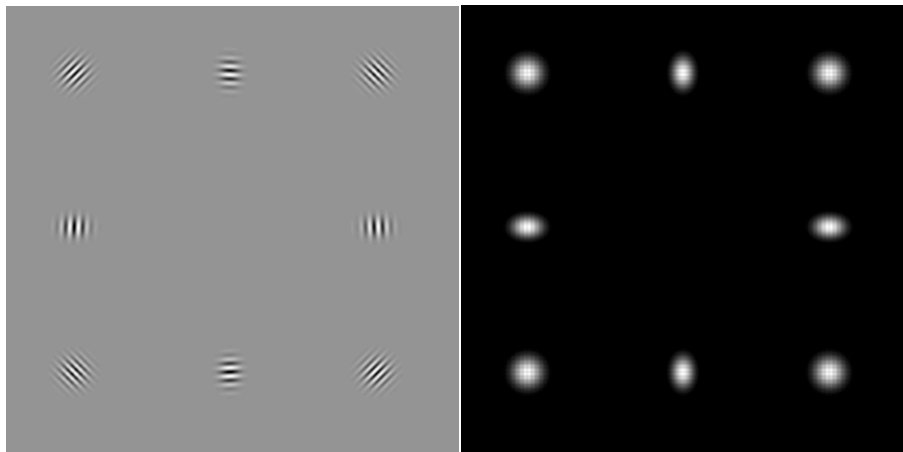
Figure 5.4: Directional selective Gabor-like wavelets. Left: Real component of the complex wavelets. Right: Envelopes of the complex wavelets. We note that the diagonally placed wavelets are identical, they are repeated to balance the representation.

where $\sigma = \sqrt{\alpha + 1/12}$. Along with (5.24), it is then straightforward to deduce (5.18).

Figure 5.4 shows these *Gabor-like wavelets* for $\alpha = 6$ and $\tau = 0$. The ensemble shows the modulus $|\Psi_k(x;6,0)|$ and the real component $\mathbf{Re}[\Psi_k(x;6,0)]$ of the six complex wavelets. The former clearly shows the pulsations of the directional plane waves, whereas the latter shows the Gaussian envelopes.

5.5 Implementation

We now discuss the implementation of the discrete wavelet transforms corresponding to the Hilbert pairs in (5.17). As before, we take some sufficiently large α, and arbitrarily set $\tau = 0$. Our eventual goal is to compute the coefficients of the Gabor coefficients at dyadic resolutions for every $1 \le l \le 6$,

$$c_{l,i}[k] = \int f(x)\Psi_l(2^{-i}x - k)\,dx \qquad (i \in \mathbf{Z}, k \in \mathbf{Z}^2).$$

This results in a multiresolution, directional analysis of $f(x)$ in terms of the sequence of projections $c_{l,i}[k]$.

We recall from Chapter 4 that this involves four multiresolutions. Since each of the multiresolutions are separable, the associated filters can be implemented efficiently through successive filtering along either dimension.

5.5.1   Pre-filter

The signal is projected onto each of the four approximation spaces of the form $\mathbf{V}(\varphi_1) \otimes \mathbf{V}(\varphi_2)$ before initiating the wavelet decompositions. The orthogonal projection is achieved in a separable fashion using an appropriate pre-filter along each dimension. In particular, if $f[k]$ are the samples of a bandlimited signal, then the projection coefficients are given by

$$c_0^{\mathrm{LL}}[k] = (f * p)[k]$$

where the pre-filter $p[k] = p[k_1, k_2]$ is specified by

$$\sum p[k_1, k_2] \exp\left[-j(k_1\omega_1 + k_2\omega_2)\right] = \hat{\phi}_1(\omega_1)^* \hat{\phi}_2(\omega_2)^*.$$

This gives the projections

$$c_{0,n}^{\mathrm{LL}}[k] = (f * p_n)[k] \qquad (1 \le n \le 4)$$

corresponding to the pre-filters $p_1[k], \ldots, p_4[k]$.

5.5.2   Analysis

We consider the implementation aspects for finite periodic data $\mathbf{f}$ of size $M \times N$, e.g., an image with periodic boundary conditions. Let us denote the low-low (LL), low-high (LH), high-low (HL) and high-high (HH) sub-bands of the four decompositions at resolutions $i = 1, \ldots, J$ by

$$c_{i,n}^{\mathrm{LL}}[k], \; c_{i,n}^{\mathrm{LH}}[k], \; c_{i,n}^{\mathrm{HL}}[k] \; \text{ and } \; c_{i,n}^{\mathrm{HH}}[k] \qquad (1 \le n \le 4).$$

The sub-bands $c_{0,n}^{\mathrm{LL}}$ are simply the pre-filtered signals

$$c_{0,n}^{\mathrm{LL}} = \mathbf{P}_n \mathbf{f} \qquad (1 \le n \le 4) \tag{5.20}$$

where $\mathbf{P}_n$ are the (block) circulant matrices associated with the separable pre-filters.

The discrete wavelet transform uses 4 separable discrete wavelet transforms with different filters applied along the two directions (cf. Table 5.1). Let $\mathbf{U}_n^{\mathrm{LL}}, \ldots, \mathbf{U}_n^{\mathrm{HH}}$ denote the composition of the analysis matrix (with appropriate filters along the two directions), and the standard downsampling matrix for the subbands LL, ..., HH and for each $1 \le n \le 4$. The subbands at coarser scales $i = 1, \ldots, J$ are given by

$$\left. \begin{aligned}
c_{i,n}^{\mathrm{LL}} &= \mathbf{U}_n^{\mathrm{LL}} \, c_{i-1,n}^{\mathrm{LL}} \\
c_{i,n}^{\mathrm{LH}} &= \mathbf{U}_n^{\mathrm{LH}} \, c_{i-1,n}^{\mathrm{LH}} \\
c_{i,n}^{\mathrm{HL}} &= \mathbf{U}_n^{\mathrm{HL}} \, c_{i-1,n}^{\mathrm{HL}} \\
c_{i,n}^{\mathrm{HH}} &= \mathbf{U}_n^{\mathrm{HH}} \, c_{i-1,n}^{\mathrm{HH}}
\end{aligned} \right\} \qquad (1 \le n \le 4). \tag{5.21}$$

Table 5.1: Components of the separable wavelet filters

n	Initial spaces	Analysis filters		Synthesis filters	
		x-direction	y-direction	x-direction	y-direction
1	$V(\varphi) \otimes V(\varphi)$	$\tilde{h}, \tilde{g}$	$\tilde{h}, \tilde{g}$	h, g	h, g
2	$V(\varphi) \otimes V(\varphi')$	$\tilde{h}, \tilde{g}$	$\tilde{h}', \tilde{g}'$	h, g	h', g'
3	$V(\varphi') \otimes V(\varphi)$	$\tilde{h}', \tilde{g}'$	$\tilde{h}, \tilde{g}$	h', g'	h, g
4	$V(\varphi') \otimes V(\varphi')$	$\tilde{h}', \tilde{g}'$	$\tilde{h}', \tilde{g}'$	h', g'	h', g'

Consider the following compound images obtained by the concatenation of appropriate sub-bands:

$$\zeta_i = \left(c_{i,1}^{\mathrm{HL}}, c_{i,2}^{\mathrm{HL}}, c_{i,1}^{\mathrm{LH}}, c_{i,3}^{\mathrm{LH}}, c_{i,1}^{\mathrm{HH}}, c_{i,4}^{\mathrm{HH}} \right),$$

$$\xi_i = \left(c_{i,3}^{\mathrm{HL}}, c_{i,4}^{\mathrm{HL}}, c_{i,2}^{\mathrm{LH}}, c_{i,4}^{\mathrm{LH}}, c_{i,2}^{\mathrm{HH}}, c_{i,3}^{\mathrm{HH}} \right).$$

These are particular permutations of the 12 high pass sub-bands,

$$\Pi : \left\{ c_i^{\mathrm{LH}}(n), c_i^{\mathrm{HL}}(n), c_i^{\mathrm{HH}}(n) \right\}_{1 \leq n \leq 4} \mapsto (\zeta_i, \xi_i). \tag{5.22}$$

The complex wavelet sub-bands are then given by

$$\mathbf{w}_i = \mathbf{\Lambda_R} \zeta_i + j \mathbf{\Lambda_I} \xi_i \qquad (1 \leq i \leq J)$$

where

$$\mathbf{\Lambda_R} = \frac{1}{\sqrt{2}} \begin{pmatrix} \sqrt{2}\mathbf{I} & 0 & 0 & 0 & 0 & 0 \\ 0 & \sqrt{2}\mathbf{I} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{2}\mathbf{I} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{2}\mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} & -\mathbf{I} \\ 0 & 0 & 0 & 0 & \mathbf{I} & \mathbf{I} \end{pmatrix}$$

and

$$\mathbf{\Lambda_I} = \frac{1}{\sqrt{2}} \begin{pmatrix} \sqrt{2}\mathbf{I} & 0 & 0 & 0 & 0 & 0 \\ 0 & \sqrt{2}\mathbf{I} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{2}\mathbf{I} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{2}\mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} & \mathbf{I} \\ 0 & 0 & 0 & 0 & \mathbf{I} & -\mathbf{I} \end{pmatrix}.$$

Note that each $\mathbf{w}_i = (\mathbf{w}_i^1, \ldots, \mathbf{w}_i^6)$ is a compound image consisting of the six directional sub-bands.

In short, the transform can be written as

$$T : \mathbf{f} \mapsto (c_{J,1}^{\mathrm{LL}}, \ldots, c_{J,4}^{\mathrm{LL}}, \mathbf{w}_1, \ldots, \mathbf{w}_J). \tag{5.23}$$

It uses the following transformations in order: four pre-filters, four discrete wavelet transforms, permutations of the wavelet sub-bands, and two the orthonormal transformations $\Lambda_{\mathbf{R}}$ and $\Lambda_{\mathbf{I}}$. Figure 5.5 provides a schematic of these sequence of transformations.
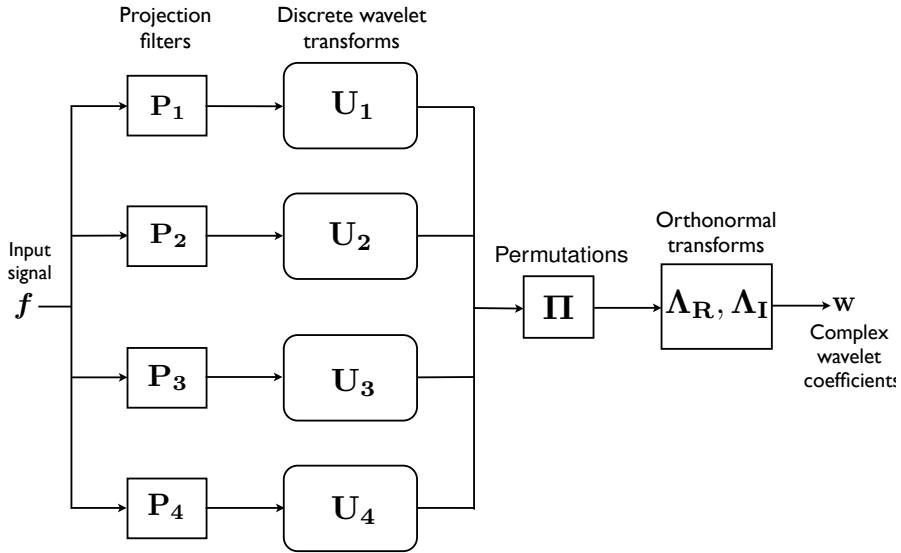
Figure 5.5: Block diagram of the two-dimensional Gabor-like transform.

5.5.3   Reconstruction

Note the Π given by (5.22) is trivially invertible. Also, $\Lambda_{\mathbf{R}}$ and $\Lambda_{\mathbf{I}}$ are orthonormal, with corresponding inverses given by $\Lambda_{\mathbf{R}}^{T}$ and $\Lambda_{\mathbf{I}}^{T}$. Starting from the wavelet sub-bands, we first recover the compound images,

$$\zeta_i = \Lambda_{\mathbf{R}}^{T}\mathbf{Re}(\mathbf{w}_i) \quad \text{and} \quad \xi_i = \Lambda_{\mathbf{I}}^{T}\mathbf{Im}(\mathbf{w}_i).$$

The 12 high pass sub-bands are retrieved from these compound images vectors at every scale $i = 1, \ldots, J$ using the permutation Π^{-1}.

Let $\mathbf{V}_n^{\mathrm{LL}}, \ldots, \mathbf{V}_n^{\mathrm{HH}}$ denote the composition of the upsampling matrix and the separable synthesis matrix for the $\mathrm{LL}, \ldots, \mathrm{HH}$ sub-bands (specified in Table 5.1) for each $1 \leq n \leq 4$. The signals $c_{0,1}^{\mathrm{LL}}, \ldots, c_{0,4}^{\mathrm{LL}}$ are reconstructed using the recursion

$$c_{i,n}^{\mathrm{LL}} = \mathbf{V}_n^{\mathrm{LL}}c_{i+1,n}^{\mathrm{LL}} + \mathbf{V}_n^{\mathrm{LH}}c_{i+1,n}^{\mathrm{LH}} + \mathbf{V}_n^{\mathrm{HL}}c_{i+1,n}^{\mathrm{HL}} + \mathbf{V}_n^{\mathrm{HH}}c_{i+1,n}^{\mathrm{HH}} \quad (i = J-1, \ldots, 0).$$

The input signal samples are finally recovered using the averaging

$$\mathbf{f} = \frac{1}{4} \sum_{n=1}^{4} \mathbf{P}_n^{-1} c_{0,n}^{\mathrm{LL}}$$

Figure 5.6 shows the magnitude of the complex wavelet sub-bands obtained by applying our Gabor-like transform to a synthetic and a natural image. The wavelet sub-bands corresponding to the synthetic image, with directional edges along $0, \pi/4, \pi/2$ and $3\pi/4$, highlight the directional-selectivity of the transform. The simulation was carried out in MATLAB 7.5 on a Macintosh 2.66 GHz Intel dual-core system. The filter bank operations are implemented in a separable fashion in the frequency domain. The average execution time for one-level wavelet analysis and reconstruction (including pre- and post-filtering) of a 512×512 image is 1.2 seconds, and the reconstruction error is of the order of 10^{-16}.

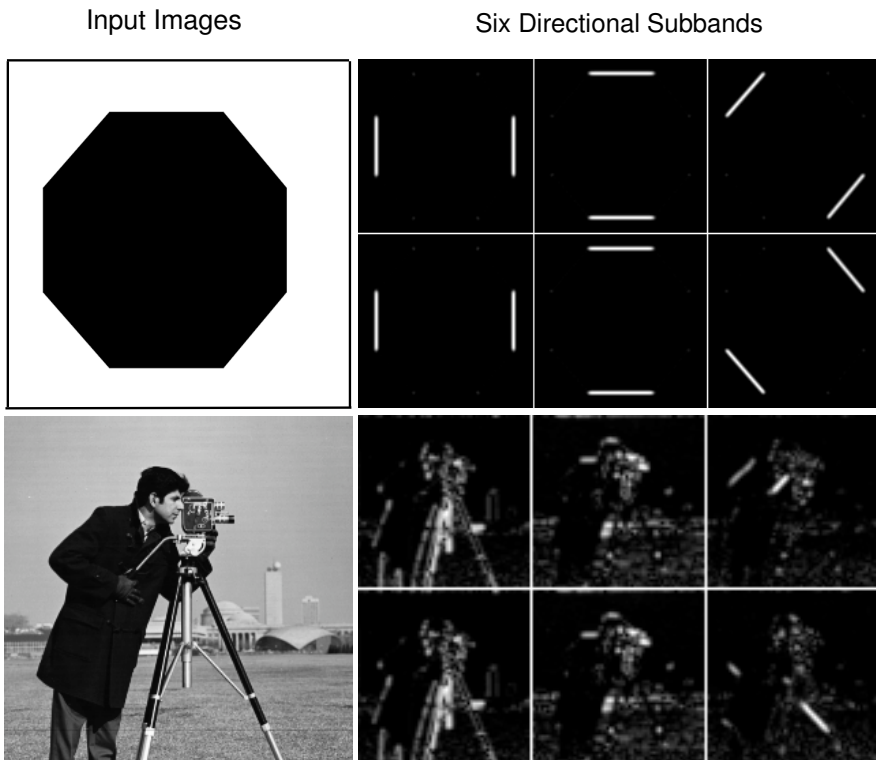Input Images   Six Directional Subbands

Figure 5.6: Directional decomposition of a synthetic image (Octagon) and a natural image (Cameraman) using the Gabor-like transform.

Appendix A

To establish (5.9), all we need to show is that

$$\lim_{\alpha \longrightarrow \infty} \psi_\tau^\alpha(x) \sim \frac{2M^{\alpha+1}\sigma}{\sqrt{2\pi(\alpha+1)}} \exp\left[- \frac{\sigma^2}{2(\alpha+1)}(x - 1/2)^2 \right] \cos\left(\omega_0(x - 1/2) - \pi\tau\right)$$

$$(5.24)$$

where M, ω_0 and σ are appropriate parameters (to be determined).

We proceed by taking the Fourier transform of (5.2),

$$\hat{\psi}_\tau^\alpha(2\omega) = G(e^{j\omega})\hat{\beta}_\tau^\alpha(\omega).$$

It can be verified that

$$\pm j\omega = |\omega| \exp\left(\pm j\pi\tau \mathrm{sgn}(\omega) \right)$$

and

$$1 - \exp(\pm j\omega) = 2|\sin(\omega/2)| \exp\left(\pm \omega/2 \mp \pi/2 \right).$$

We can thus write (5.1) as

$$\hat{\beta}_\tau^\alpha(\omega) = |\mathrm{sinc}(\omega/2)|^{\alpha+1} \exp\left(j\pi\tau - j\tau\omega - j\pi\tau\mathrm{sgn}(\omega) \right).$$

Similarly, it can be show that

$$H(e^{j\omega}) = |\cos(\omega/2)|^{\alpha+1} \exp(-j\tau\omega)$$

and

$$G(e^{j\omega}) = |\sin(\omega/2)|^{\alpha+1} \sum_{n \in \mathbf{Z}} \left[\mathrm{sinc}\left(\frac{\omega - (2n+1)\pi}{2} \right) \right]^{2\alpha+2} \exp\left(j\omega + j\tau\omega - j\pi\tau \right).$$

Combining them, we can write

$$\hat{\psi}_\tau^\alpha(\omega) = \exp\left(j\omega/2 - j\pi\tau\mathrm{sgn}(\omega) \right) \sum_{k \in \mathbf{Z}} s_k(\omega)^{\alpha+1} \qquad (5.25)$$

where

$$s_k(\omega) = |\sin(\omega/4)\mathrm{sinc}(\omega/4)| \left[\mathrm{sinc}\left(\frac{\omega - 2(2k+1)\pi}{4} \right) \right]^2.$$

We now study the behavior of the series

$$\sum_{k \in \mathbf{Z}} s_n(\omega)^{\alpha+1} = s_0(\omega)^{\alpha+1} + s_{-1}(\omega)^{\alpha+1} + \sum_{k=1}^{\infty} \left[s_{2k+1}(\omega)^{\alpha+1} + s_{2k+1}(-\omega)^{\alpha+1} \right]$$

for large α. We claim that

$$\lim_{\alpha \to \infty} \left[\frac{1}{M} s_0 \left(\frac{\omega}{\sqrt{\alpha+1}} + \omega_0 \right) \right]^{\alpha+1} = \exp \left(-\frac{\omega^2}{2\sigma^2} \right), \qquad (5.26)$$

$$\lim_{\alpha \to \infty} \left[\frac{1}{M} s_{-1} \left(\frac{\omega}{\sqrt{\alpha+1}} - \omega_0 \right) \right]^{\alpha+1} = \exp \left(-\frac{\omega^2}{2\sigma^2} \right), \qquad (5.27)$$

and

$$\lim_{\alpha \to \infty} \sum_{k=1}^{\infty} \left[\frac{1}{M} s_{2k+1} \left(\frac{\omega}{\sqrt{\alpha+1}} \right) \right]^{\alpha+1} + \left[\frac{1}{M} s_{2k+1} \left(-\frac{\omega}{\sqrt{\alpha+1}} \right) \right]^{\alpha+1} = 0. \quad (5.28)$$

Based on these, we conclude from (5.25) that

$$\lim_{\alpha \to \infty} \left| \hat{\psi}_\tau^\alpha \left(\frac{\omega}{\sqrt{\alpha+1}} \pm \omega_0 \right) \right| = \exp \left(-\frac{\omega^2}{2\sigma^2} \right).$$

We now put back the phase terms, and undo the centering and normalization, to obtain

$$\lim_{\alpha \to \infty} \hat{\psi}_\tau^\alpha(\omega) \sim M^{\alpha+1} e^{j\omega/2} \left[e^{-j\pi\tau} \exp \left(-\frac{(\omega-\omega_0)^2}{2\sigma^2} \right) + e^{j\pi\tau} \exp \left(-\frac{(\omega+\omega_0)^2}{2\sigma^2} \right) \right].$$

We claim that the inverse Fourier transform of both sides are again asymptotically equivalent. This gives us (5.24).

We now establish the convergences in (5.26), (5.27) and (5.28). For (5.26) and (5.27), we note that if $f(x)$ is continuously differentiable and has a positive maximum at x_0, then

$$f(x) = f(x_0) \left[1 - \frac{(x-x_0)^2}{2\sigma^2} + o(|x-x_0|^2|) \right]$$

where $\sigma = (f(x_0)/|f''(x_0)|)^{1/2}$. This is immediate from Taylor's theorem and the fact that $f'(x_0) = 0$ and $f''(x_0) < 0$.

Now consider the functions

$$\mu_\alpha(x) = \left[\frac{1}{f(x_0)} f \left(\frac{x}{\sqrt{\alpha}} + x_0 \right) \right]^\alpha \qquad (\alpha > 0).$$

Form the previous result,

$$\log \mu_\alpha(x) = \alpha \log \left(1 - \frac{x^2}{2\alpha\sigma^2} + o(\alpha^{-1}) \right) = \left(-\frac{x^2}{2\sigma^2} + o(1) \right)$$

so that

$$\lim_{\alpha \longrightarrow \infty} \mu_\alpha(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right).$$ (5.29)

A simple computation shows that

$$s_0(\omega) = \frac{16|\sin(\omega/2)|^2}{|\omega|(\omega - 2\pi)^2}.$$

It can be verified that this has a global maximum at $\omega_0 \approx 5.14$. Therefore, setting

$$M = s_0(\omega_0) \approx 0.697 \quad \text{and} \quad \sigma = (s_0(\omega_0)/|s_0''(\omega_0)|)^{1/2} \approx 2.67$$

we at once deduce (5.26). Moreover, since $s_{-1}(\omega) = s_0(-\omega)$, (5.27) follows by symmetry.

In this regard, we note that the correct centering of $\mu_\alpha(x)$ is absolutely essential for (5.29) to hold: If x_1 is different from x_0, then

$$\lim_{\alpha \longrightarrow \infty} \left[\frac{1}{f(x_0)} f\left(\frac{x}{\sqrt{\alpha}} + x_1\right)\right]^\alpha = 0.$$

In particular, this means that

$$\lim_{\alpha \longrightarrow \infty} \left[\frac{1}{M} s_0\left(\frac{\omega}{\sqrt{\alpha+1}} - \omega_0\right)\right]^{\alpha+1} = 0.$$

Now to establish (5.28), we note that

$$\sum_{k=1}^{\infty} \left(s_{2k+1}(2\omega)^{\alpha+1} + s_{2k+1}(-2\omega)^{\alpha+1}\right) = |\mathrm{sinc}(\omega/2)\sin(\omega/2)|^{\alpha+1} r_\alpha(\pi - \omega) \quad (5.30)$$

where

$$r_\alpha(\omega) = \sum_{k=1}^{\infty} \left|\mathrm{sinc}\left(\frac{\omega - 2k\pi}{2}\right)\right|^{2\alpha+2}.$$

We claim that

$$r_\alpha(\omega) \le \sum_{k \in \mathbf{Z}} \left|\mathrm{sinc}\left(\frac{\omega - 2k\pi}{2}\right)\right|^{2\alpha+2} = 1 \quad \text{(for all } \omega\text{)} \quad (5.31)$$

and

$$r_\alpha(\omega) < \frac{4}{\pi^{2\alpha+2}}\left(1 - \frac{\omega}{2\pi}\right)^{-2\alpha-2} \quad (0 < \omega < \pi). \quad (5.32)$$

The bound in (5.31) can be derived using Poisson's summation formula. Indeed, the series on the right corresponds to the Fourier transform of the (integer) samples of

the symmetric B-spline of order $2\alpha + 1$. Since the Fourier transform is uniformly bounded by the ℓ^1 norm of the sequence of samples, the bound is realized by noting that the ℓ^1 norm of the samples of a B-spline function (any arbitrary order) always equals unity.

On the other hand, (5.32) follows from the estimate

$$
\sum_{k=1}^{\infty} \left| \mathrm{sinc}\left(\frac{\omega - 2k\pi}{2} \right) \right|^{2\alpha+2} < 2 \sum_{k=1}^{\infty} \left| \mathrm{sinc}\left(k\pi - \frac{\omega}{2} \right) \right|^{2\alpha+2}
$$

$$
\leq \frac{2}{\pi^{2\alpha+2}} \sum_{k=1}^{\infty} \left(k - \frac{\omega}{2\pi} \right)^{-2\alpha-2}
$$

$$
< \frac{2}{\pi^{2\alpha+2}} \left[\left(1 - \frac{\omega}{2\pi} \right)^{-2\alpha-2} + \int_{1}^{\infty} \left(x - \frac{\omega}{2\pi} \right)^{-2\alpha-2} dx \right]
$$

$$
< \frac{4}{\pi^{2\alpha+2}} \left(1 - \frac{\omega}{2\pi} \right)^{-2\alpha-2}
$$

which holds for all $0 < \omega < \pi$.

From (5.30) and (5.31), it is seen that

$$
\sum_{k=1}^{\infty} \left[\frac{1}{M} s_{2k+1}\left(\frac{2\omega}{\sqrt{\alpha+1}} \right) \right]^{\alpha+1} + \left[\frac{1}{M} s_{2k+1}\left(-\frac{2\omega}{\sqrt{\alpha+1}} \right) \right]^{\alpha+1} \leq \left| \frac{2\sqrt{\alpha+1}}{M\omega} \right|^{\alpha+1}.
$$

Therefore, when $\omega > \pi\sqrt{\alpha+1}$, the series is bounded by $(2/M\pi)^{\alpha+1}$. This clearly goes to zero as $\alpha \longrightarrow \infty$.

On the other hand, when $0 < \omega < \pi\sqrt{\alpha+1}$, the absolute value of $\pi - \omega/\sqrt{\alpha+1}$ is less than π. Hence, from (5.30) and (5.32), we obtain that

$$
\sum_{k=1}^{\infty} \left[\frac{1}{M} s_{2k+1}\left(\frac{2\omega}{\sqrt{\alpha+1}} \right) \right]^{\alpha+1} + \left[\frac{1}{M} s_{2k+1}\left(-\frac{2\omega}{\sqrt{\alpha+1}} \right) \right]^{\alpha+1} \leq 4\left(\frac{2}{M\pi} \right)^{2\alpha+2}.
$$

This again is bounded by $4(2/M\pi)^{2\alpha+2}$ which goes to zero as $\alpha \longrightarrow \infty$.

Combining these two cases and using symmetry, we conclude that the series converges to zero as $\alpha \longrightarrow \infty$ for all ω. This establishes (5.28).

Appendix B

We give a sketch of the derivation of (5.19). Fix some τ_0, and consider the normalized B-splines

$$
\varphi_\alpha(x) = \sigma \beta_\tau^\alpha(\sigma x) \qquad (\alpha > 0) \tag{5.33}
$$

where

$$
\sigma = \sqrt{(\alpha+1)/12} \quad \text{and} \quad \tau = \sigma\tau_0.
$$

It is sufficient to show that

$$\lim_{\alpha \longrightarrow \infty} \varphi_\alpha(x) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}(x - \tau_0)^2\right]. \qquad (5.34)$$

By undoing the normalization in (5.33), we then get (5.19).

Note that, by Taylor's theorem,

$$\left(\frac{1 - e^{-j\omega}}{j\omega}\right) = e^{-j\omega/2}\left(1 - \frac{\omega^2}{24} + o(\omega^2)\right).$$

Since $\log(1 + z) = z + o(z)$, we have from (5.1),

$$\log \hat{\beta}_\tau^\alpha(\omega) = p \log\left(\frac{1 - e^{-j\omega}}{j\omega}\right) + q \log\left(\frac{1 - e^{j\omega}}{-j\omega}\right)$$

$$= p\left(-\frac{j\omega}{2} - \frac{\omega^2}{24} + o(\omega^2)\right) + q\left(\frac{j\omega}{2} - \frac{\omega^2}{24} + o(\omega^2)\right)$$

$$= -j\omega\tau - \frac{\alpha + 1}{24}\omega^2 + (\alpha + 1)o(\omega^2).$$

Therefore, by the scaling property of the Fouier transform,

$$\log \hat{\varphi}_\alpha(\omega) = \log \hat{\beta}_{\tau(\alpha)}^\alpha\left(\frac{\omega}{\sigma_\alpha}\right) = -j\tau_0\omega - \frac{\omega^2}{2} + o(1).$$

In particular,

$$\lim_{\alpha \longrightarrow \infty} \log \hat{\varphi}_\alpha(\omega) = -j\tau_0\omega - \frac{\omega^2}{2}.$$

By the continuity of the logarithm, we conclude that

$$\lim_{\alpha \longrightarrow \infty} \hat{\varphi}_\alpha(\omega) = \exp\left(-j\tau_0 - \frac{\omega^2}{2}\right).$$

Exchanging the Fourier transform and the limit[2], we get (8.34).

Appendix C

The natural domain for discrete periodic signals of length N is $\mathbf{Z}_N$, the additive group of integers modulo N. The implementation of discrete wavelet transforms require the specification of two basic operations on $\mathbf{Z}_N$, namely that of convolution

---

[2]This is the technical part. One possible way of establishing norm convergence, say $\mathbf{L}^2$ convergence, is to extend the pointwise convergence in Fourier domain to $\mathbf{L}^2$ convergence, and then use Parseval's theorem to get $\mathbf{L}^2$ convergence in space domain.

and (dyadic) upsampling and downsampling. The downsampling operation, of course, requires the original group to be of even size.

We implement the pair of discrete wavelet transforms corresponding to the real and imaginary components of the complex wavelet in the Fourier domain.

Let $\omega_N = \exp(j2\pi/N)$. We recall that the Fourier transform of a function $f(k)$ on $\mathbf{Z}_N$ is again a function $\hat{f}(n)$ on $\mathbf{Z}_N$ given by

$$\hat{f}(n) = \sum_{k \in \mathbf{Z}_N} f(k)\omega_N^{kn} \qquad (0 \le n < N).$$

Finite periodic filters. The discrete filters (the refinement filter and the wavelet filter) involved in the Gabor-like wavelet transform are initially defined on $\mathbf{Z}$ and not on $\mathbf{Z}_N$. To specify the corresponding filters on $\mathbf{Z}_N$, we use their continuously-defined Fourier transforms.

Consider a typical filter given by the sequence $s[k]$. Its Fourier transform is

$$S(e^{j\theta}) = \sum_{n \in \mathbf{Z}} s[k]e^{jn\theta} \qquad (0 \le \theta \le 2\pi).$$

To specify the associated filter on $\mathbf{Z}_N$, we proceed by uniformly sampling $S(e^{j\theta})$ on $[0, 2\pi]$, that is, by setting

$$S_N(n) = S(\omega_N^n) \qquad (0 \le n < N).$$

This gives us the function $S_N(n)$ and a corresponding filter $s_N(n)$, both defined on $\mathbf{Z}_N$, where

$$S_N(n) = \sum_{k \in \mathbf{Z}_N} s_N(k)\omega_N^{kn} \qquad (0 \le n < N).$$

The function $s_N(n)$ is in fact the N-periodic version of the sequence $s[k]$,

$$s_N(n) = \sum_{k \in \mathbf{Z}_N} s[n + kN] \qquad (n \in \mathbf{Z}_N).$$

Using this periodic filter, we can define the convolution

$$(s_N * f)(n) = \sum_{k \in \mathbf{Z}_N} s_N(k)f(n - k) \qquad (n \in \mathbf{Z}_N).$$

In terms of the Fourier transform,

$$\widehat{(s_N * f)}(n) = S_N(n)\hat{f}(n).$$

Combined convolution and sampling using FFT. Note that the operations of upsampling and downsampling (of periodic functions) involve two groups of different sizes, namely, $\mathbf{Z}_N$ and $\mathbf{Z}_{2N}$. The upsampling operation takes a function $f(k)$ on $\mathbf{Z}_N$ into a function $u(k)$ on $\mathbf{Z}_{2N}$ given by

$$u(2k) = f(k), \qquad u(2k + 1) = 0 \qquad (0 \le k < N).$$

The downsampling operation is given by

$$d(k) = f(2k) \qquad (0 \leq k < N).$$

We efficiently realize the above convolution and dyadic sampling using the fast Fourier transform (FFT). For the sampling, we use the following formulae:

$$\hat{u}(n) = \begin{cases} \hat{f}(n) & \text{for } 0 \leq n < N \\ \hat{f}(n-N) & \text{for } N \leq n < 2N. \end{cases}$$

and

$$\hat{d}(n) = \frac{1}{2}\left[\hat{f}(n) + \hat{f}(n+N/2)\right] \qquad (0 \leq n < N/2).$$

Note that $\hat{u}(n)$ is defined on a group of size $2N$, while $\hat{d}(n)$ is defined on group of size $N/2$.

The basic routines for the discrete wavelet transform (convolution and sampling) are summarized in Algorithms 1 and 2.

Algorithm 1 Downsampling of $(s_N * f)(n)$

---

1. Input: $f(k)$ of even length N, and $S_N(k)$.
2. Compute $\hat{f}(n)$ using FFT.
3. For $0 \leq n < N/2$, set

$$y(n) \longleftarrow \tfrac{1}{2}\left[\hat{f}(n)S_N(n) + S_N(n+N/2)\hat{f}(n+N/2)\right].$$

3. Output: inverse-FFT of $y(n)$.

Algorithm 2 Upsampling of $f(k)$ followed by convolution with $s_{2N}(k)$

---

1. Input: $f(k)$ of length N, and $S_{2N}(k)$.
2. Compute $\hat{f}(n)$ using FFT.
3. For $0 \leq n < N$, set $y(n) \longleftarrow S_{2N}(n)\hat{f}(n)$.
4. For $N \leq n < 2N$, set $y(n) \longleftarrow S_{2N}(n)\hat{f}(n-N)$.
4. Output: inverse-FFT of $y(n)$.

Chapter 6

Shiftability of the Gabor-like wavelet transforms

Abstract — In this chapter, we provide an interpretation of the amplitude and phase information obtained from the Gabor-like transform introduced in Chapter 5. This interpretation is particularly relevant when we reconstruct the signal from the redundant wavelet coefficients, given that there exists several non-unique ways of doing so[1].

6.1 Introduction

THe developments in this chapter are based on the analogy with Fourier analysis which (among other things) provides a means of encoding the relative location of information in signals. We recall that the Fourier expansion of a finite-energy signal $f(x)$ on $[0, L]$ is given by

$$
\begin{aligned}
f(x) = a_0 &+ a_1 \cos(\omega_0 x) + a_2 \cos(2\omega_0 x) + \cdots \\
&+ b_1 \sin(\omega_0 x) + b_2 \sin(2\omega_0 x) + \cdots
\end{aligned} \tag{6.1}
$$

where $\omega_0 = 2\pi/L$ is the fundamental frequency, and $a_0, a_1, a_2, \ldots$, and $b_1, b_2, \ldots$ are the proportions of the even and odd harmonics. Let us introduce the complex coefficients $c_n = a_n - j b_n$ and express them in the polar form $c_n = |c_n| \exp(j\phi_n)$,

---

[1]This chapter is based on the article [33]: K. N. Chaudhury, M. Unser, "On the shiftability of dual-tree complex wavelet transforms," IEEE Transactions on Signal Processing, vol. 58, no. 1, pp. 221-232, January 2010.

where $0 \leqslant \phi_n < 2\pi$. We can then write (6.1) as

$$f(x) = \sum_{n=0}^{\infty} |c_n| \left(\cos \phi_n \cos(n\omega_0 x) - \sin \phi_n \sin(n\omega_0 x) \right)$$

$$= \sum_{n=0}^{\infty} |c_n| \varphi_n(x + \tau_n) \tag{6.2}$$

where

$$\varphi_n(x) = \cos(n\omega_0 x) \quad \text{and} \quad \tau_n = \phi_n / n\omega_0.$$

In this form, the *shift* τ_n specifies the displacement of the reference sinusoid $\varphi_n(x)$ relative to its fundamental period $[0, L/n]$. The fundamental aspect of (6.2) is that it provides a characterization of the modulus and phase information using the following optimality criterion: From the orthogonality of the harmonic series, one can show that τ_n corresponds to that shift which maximizes $|\langle f(\cdot), \varphi_n(\cdot + \tau) \rangle|$, the correlation between the signal and $\varphi_n(x)$. The amplitude $|c_n|$ is simply the maximum correlation,

$$|c_n| = \max_{\tau} |\langle f(\cdot), \varphi_n(\cdot + \tau) \rangle|.$$

Fourier analysis involves "global" basis functions and proves to be less efficient for approximating signals with isolated singularities, e.g., signals such as natural images which are piecewise-polynomial. The wavelet representation is known to be more efficient in this case [14]. This is attributed to their characteristic *vanishing moment* property. Added to this is the multiresolution structure which allows one to localize signal discontinuities that live at different spatial resolutions.

Complex wavelets, particularly the ones obtained by combining non-redundant wavelet bases, provide an attractive means of recovering the phase information. The phase relation between the components of the complex wavelet is used to encode the relative signal displacement. Our present interest is in the case where the components form a Hilbert pair.

6.2   Shiftable representation

We make some basic observations about (6.2). Note that

$$c_n = \int_0^L f(x) \left[e^{jn\omega_0 x} \right]^* dx \tag{6.3}$$

where $e^{jn\omega_0 x}$ is the primitive analytic function $e^{jn\omega_0 x} = \varphi_n(x) + j \mathcal{H} \varphi_n(x)$. Next, we use the phase-shift property of the fractional Hilbert transforms (fHT) introduced in Chapter 2, to express the "reconstruction" functions in (6.2) as

$$\varphi_n(x + \tau_n) = \mathcal{H}_{\tau_n} \varphi_n(x)$$

where we redefine τ_n to be $\tau_n = \phi_n/\pi$. We can then write (6.2) as

$$f(x) = \sum_{n=0}^{\infty} |c_n| \, \mathcal{H}_{\tau_n} \varphi_n(x). \tag{6.4}$$

We now show how (6.4) can be generalized by replacing the global waveforms $\varphi_n(x)$ by localized wavelets. We recall from Chapter 5 that the Gabor-like transform involves two wavelet bases

$$(\psi_{i,k}) \quad \text{and} \quad (\psi'_{i,k}) \qquad (i \in \mathbf{Z}, k \in \mathbf{Z}).$$

These are obtained though the dilations and translations of mother wavelets $\psi(x)$ and $\psi'(x)$, where

$$\psi'(x) = \mathcal{H} \psi(x).$$

The representation of signal $f(x)$ in $L^2(\mathbf{R})$ in terms of these bases is given by

$$f(x) = \begin{cases} \displaystyle\sum_{(i,k)\in\mathbf{Z}^2} a_i[k]\psi_{i,k}(x) \\ \displaystyle\sum_{(i,k)\in\mathbf{Z}^2} b_i[k]\psi'_{i,k}(x). \end{cases} \tag{6.5}$$

The coefficients $a_i[k]$ and $b_i[k]$ are given by

$$a_i[k] = \langle f, \tilde{\psi}_{i,k} \rangle \quad \text{and} \quad b_i[k] = \langle f, \tilde{\psi}'_{i,k} \rangle.$$

By symmetry, the dual wavelet bases are generated through the dilations and translations of the dual wavelets that form a Hilbert pair,

$$\tilde{\psi}'(x) = \mathcal{H} \tilde{\psi}(x).$$

By analogy with the Fourier representation, we introduce the complex coefficients

$$c_i[k] = \frac{1}{2}\big(a_i[k] - j\,b_i[k]\big) = |c_i[k]| \exp(j\phi_i[k]).$$

Introducing the analytic wavelet

$$\tilde{\Psi}(x) = \frac{1}{2}\Big(\tilde{\psi}(x) + j\,\tilde{\psi}'(x)\Big)$$

we can write this as

$$c_i[k] = \int f(x)\tilde{\Psi}^*_{i,k} \, dx.$$
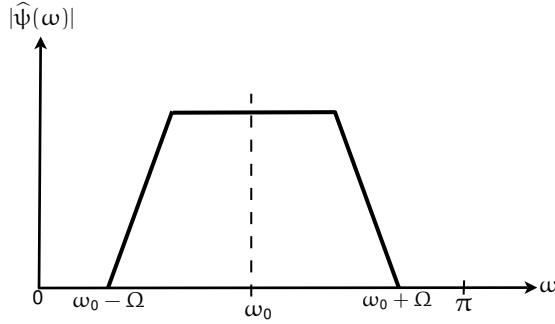
This is analogous to (6.3).

Figure 6.1: Ideal spectrum of a modulated wavelet. The spectrum has passbands over $\omega_0 - \Omega < |\omega| < \omega_0 + \Omega$ with local axes of symmetry at $\omega = \pm\omega_0$.

Clearly, $c_i[k]$ is a redundant representation of $f(x)$. We choose to reconstruct the signal by averaging the individual reconstructions in (6.5). This results in the representation

$$f(x) = \frac{1}{2} \sum_{(i,k)\in\mathbf{Z}^2} \left[a_i[k]\psi_{i,k}(x) + b_i[k]\psi'_{i,k}(x) \right]$$

$$= \sum_{(i,k)\in\mathbf{Z}^2} |c_i[k]| \, \mathcal{H}_{\phi_i[k]/\pi} \left[\psi_{i,k}(x) \right].$$

Let us denote

$$\psi(x;\tau) = \mathcal{H}_\tau \psi(x) \qquad (\tau \in \mathbf{R})$$

and use $\mathrm{Dil}_i\mathrm{Trans}_k[f(x)]$ to represent $f_{i,k}(x)$. Using the dilation and translation invariance of $\mathcal{H}_\tau$, we can write

$$f(x) = \sum_{(i,k)\in\mathbf{Z}^2} |c_i[k]| \, \mathrm{Dil}_i\mathrm{Trans}_k \left[\psi(x;\tau_i[k]) \right]. \tag{6.6}$$

This is analogues to (6.4) in that it gives the signal representation in terms of the modulus and phase information. The coefficients in this case are obtained by projecting the signal on to the dilated-translated copies of the analytic wavelet. The multiresolution amplitude-phase representation in (6.6) provides an important insight into the signal transformation

$$f(x) \mapsto \left\{ \left(|c_i[k]|, \tau_i[k] \right) \right\}_{(i,k)\in\mathbf{Z}^2}.$$

Note that the shifted wavelets $\psi(x;\tau_i[k])$ in (6.6) play a role analogous to the phase-shifted sinusoids $\varphi_n(x+\tau_n)$ in (6.2). This analogy provides a formal interpretation

of the amplitude-phase factors: The modulus $|c_i[k]|$ indicates the strength of the local signal correlation at scale i and position $2^i k$. On the other had, the relative signal displacement is encoded in the shift $\tau_i[k]$, which corresponds to the most "appropriate" waveform within the family $\mathscr{H}_\tau \psi_{i,k}(x)$.

6.3 Modulated wavelets

We now show that the functions $\psi(x; \tau_i[k])$ can be explicitly characterized for certain classes of wavelets. This provides further insight into (6.6).

A wavelet is a bandpass function by construction. We consider the particular class of *modulated wavelets* of the form

$$\psi(x) = \varphi(x)\cos(\omega_0 x + \xi_0).$$

We assume that the localization window $\varphi(x)$ is bandlimited to $-\Omega \leq \omega \leq \Omega$, and where $|\omega_0| > \Omega$ (cf. Figure 6.2). Under these assumptions, it was shown in Chapter 2 that

$$\mathscr{H}_\tau\left[\varphi(x)\cos(\omega_0 x + \xi_0)\right] = \varphi(x)\cos(\omega_0 x + \xi_0 + \pi\tau). \tag{6.7}$$

That is, the fractional Hilbert transform acts on the phase of the modulating sinusoid while preserving the lowpass window. Using (6.7), we can write (6.6) as

$$f(x) = \sum_{(i,k)\in\mathbf{Z}^2} \overbrace{\varphi_{i,k}(x)}^{\text{fixed window}} \text{Dil}_i\, \text{Trans}_k \left[\overbrace{\big|c_i[k]\big|\cos\big(\omega_0 x + \xi_0 + \pi\tau_i[k]\big)}^{\text{variable amp−phase oscillation}} \right] \tag{6.8}$$

where $\varphi_{i,k}(x) = \text{Dil}_i\, \text{Trans}_k\, \varphi(x)$ is the window at scale i and translation $2^i k$.

Note that the shift parameters in (6.8) and (6.2) play similar roles, namely, that of positioning the oscillation. We thus conclude (at least least intuitively) that the modulus and phase information in (6.8) are such that they "optimally" fit the signal transitions within a fixed localization window. We will shortly give an empirical demonstration of this principle using an example.

Two instances of modulated wavelets are as follows:

(1) **Shannon wavelet**. This is specified by

$$\psi(x) = \text{sinc}\left(\frac{x - 1/2}{2}\right)\cos\left(\frac{3\pi(x - 1/2)}{2}\right).$$

Its dilates and translates form an orthonormal wavelet basis of $\mathbf{L}^2$. Many wavelet families converge to the Shannon wavelet as the order increases [58]; e.g., the orthonormal Battle-Lemarié wavelets [59], and the interpolating Dubuc-Deslauriers wavelets [60].
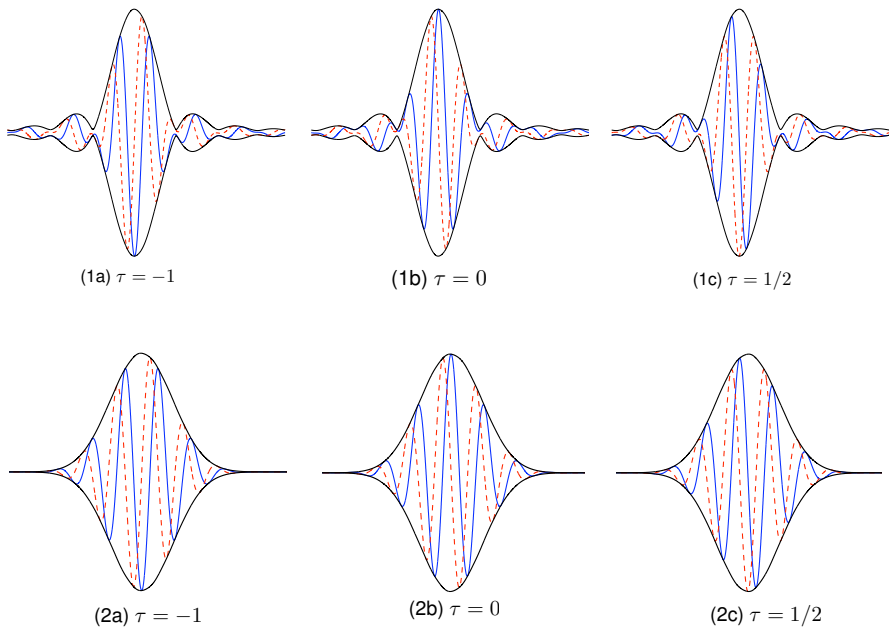
Figure 6.2: Quadrature pairs of orthonormal spline (respectively B-spline) wavelets resembling the Shannon (respectively Gabor) wavelet: Solid (blue) graph: $\mathcal{H}_\tau \psi^3(x)$; Broken (red) graph: $\mathcal{H}_{\tau+1/2}\psi^3(x)$; and Solid (black) graph: Common localization window given by $|\mathcal{H}_\tau \psi^3(x) + j\mathcal{H}_{\tau+1/2}\psi^3(x)|$.

(2) **Gabor wavelet**. The $\mathrm{sinc}(x)$ envelope of the Shannon wavelet results in an "ideal" frequency resolution, but only at the expense of poor spatial decay. As against this, wavelets modeled on the Gabor functions tend to exhibit better space-frequency localization. As in the case of the Shannon wavelet, several wavelet families closely resemble the Gabor function. It was shown in Chapter 5 the complex B-spline wavelets asymptotically converge to a Gabor function. In this case, the representation in (6.8) would correspond to the situation where the dual wavelet is used for analysis, while the Gabor wavelet is used for reconstruction.

Figure 6.2 shows quadrature pairs of Shannon-like (resp. Gabor-like) wavelets.

To empirically demonstrate the optimality of reconstruction in (6.8), we perform the following simple experiment. We consider the Gabor-like B-spline wavelet $\psi(x)$ of degree 3, and a step input $f(x) = \mathrm{sign}(x - x_0)$ which has a discontinuity at x_0. The M-level decomposition of this signal using the standard discrete wavelet transform

is given by

$$f(x) = \sum_l p_l \varphi_{M,l}(x) + \sum_{i=1}^{M} \sum_k a_{i,k} \psi_{i,k}(x), \qquad (6.9)$$

where $\varphi_{M,l}(x)$ are the translates of the coarsest scaling function. On the other hand, the Gabor-like transform gives the representation

$$f(x) = \sum_l p_l \varphi_{M,l}(x) + \sum_n p'_n \varphi'_{M,n}(x) + \sum_{i=1}^{M} \sum_k |c_i[k]| \, \psi_{i,k}(x; \tau_i[k]). \qquad (6.10)$$

The idea is to demonstrate that the shifted wavelets in (6.10) have a better lock on the transition at x_0 than the wavelets in (6.9). Figure 6.3 shows the reference wavelet $\psi_{i,k}(x)$ and the shifted wavelet $\psi_{i,k}(x; \tau_i[k])$ corresponding to a specific scale $i = J$. We set the translation to $k = [2^{-J} x_0]$ which corresponds to the position of the transition at this scale. Also shown in the figures are the step input and the Gaussian-like localization window of the wavelet. The oscillation of the shifted wavelet is clearly seen to have a better lock on the transition than the reference. The magnitude of the signal correlation in either case supports this observation.

6.4 Quality indices

The shiftability of the Gabor-like transform was established based on two fundamental properties, namely,

(1) The correspondence $\psi_2(x) = \mathcal{H} \psi_1(x)$, and

(2) The modulated forms

$$\psi_1(x) = \varphi(x) \cos(\omega_0 x + \xi_0) \quad \text{and} \quad \psi_2(x) = \varphi(x) \sin(\omega_0 x + \xi_0)$$

where the support of $\hat{\varphi}(\omega)$ is restricted to the interval $[-\omega_0, \omega_0]$.

The practical challenge is the design of different flavors of wavelets that fulfill or, at least, provide close approximations of these criteria. We propose some indices for assessing the quality of the approximation.

A simple index for (1) is the correlation

$$\varrho = \max \left(\frac{\langle \psi_2, \mathcal{H} \psi_1 \rangle}{\|\psi_1\| \cdot \|\psi_2\|}, 0 \right). \qquad (6.11)$$

By the Cauchy-Schwarz inequality, $0 \leqslant \varrho \leqslant 1$ and equals unity if and only if $\psi_2(x) = \mathcal{H} \psi_1(x)$. Thus, the higher the value of ϱ, the better is the approximation.

On the other hand, note that (2) has a simple Fourier domain characterization:

$$S(\omega) = \hat{\psi}_1(\omega) + j \hat{\psi}_2(\omega) = \begin{cases} g(\omega) & 0 < \omega < \infty \\ 0 & -\infty < \omega \leqslant 0, \end{cases}$$

$$\langle f(x), \psi_{i,k}(x;0)\rangle = 21.1$$

(a)

$$\langle f(x), \psi_{i,k}(x;\tau_i[k])\rangle = 29.3$$

(b)

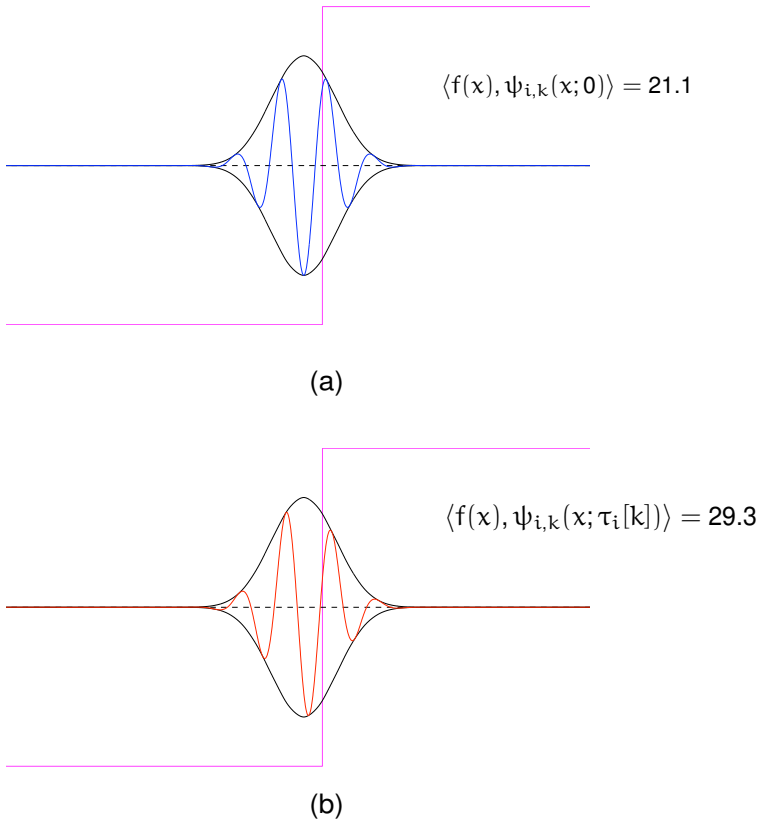**Figure 6.3:** Reconstruction wavelets for the step unit. (a) This shows the reference wavelet $\psi_{i,k}(x) = \psi_{i,k}(x;0)$ corresponding to the conventional discrete wavelet transform. (b) The shifted wavelet $\psi_{i,k}(x;\tau_i[k])$ for the Gabor-like transform is shown. The magnitudes of the signal correlation in either case clearly shows that the shifted wavelet has a better lock on the singularity.

Table 6.1: Quality indices for different classes of wavelets

Type of Dual-Tree Wavelets	ϱ	$\varkappa$
Shannon wavelets (ideal)	1	0
B-spline wavelets, degree=1	1	0.9245
B-spline wavelets, degree=3	1	0.0882
B-spline wavelets, degree=6	1	0.0373
Orthonormal spline wavelets, degree=1	1	0.9292
Orthonormal spline wavelets, degree=3	1	0.1570
Orthonormal spline wavelets, degree=6	1	0.0612
Kingsbury's wavelets (q-shift Le Gall 5/3) [27]	0.9992	0.6586

where $g(\omega)$ is complex-valued and has an axis of symmetry within its support. This suggests a index which measures the symmetry of $S(\omega)$. In particular, let

$$\varkappa = \frac{\displaystyle\int_0^\infty |S^*(\omega_0 + \omega) - S(\omega_0 - \omega)|^2 \, d\omega}{\displaystyle\int_{-\infty}^\infty |S(\omega)|^2 \, d\omega} \tag{6.12}$$

where

$$\omega_0 = \frac{\displaystyle\int_{-\infty}^\infty \omega \, |S(\omega)|^2 \, d\omega}{\displaystyle\int_{-\infty}^\infty |S(\omega)|^2 \, d\omega}.$$

Note that $0 \leq \varkappa \leq 1$. It measures the disparity between $S(\omega)$ and its reflection around the centroid, and equals zero if and only if $S(\omega)$ is symmetric (with ω_0 as the centre of symmetry). Conversely, a high value of $\varkappa$ signifies greater local asymmetry in $S(\omega)$, and hence a poor approximation of the modulation criterion.

We computed the indices ϱ and $\varkappa$ for different classes of wavelets[2]. The results are compared in Table 6.1. The spline wavelets are analytic by construction [49], and hence $\varrho = 1$ irrespective of their degree. As their degree increases, the complex B-spline (resp. orthonormal) wavelets converge to complex Gabor (resp. Shannon) wavelet. The rapid decrease in $\varkappa$ reflects the improvement in symmetry.

---

[2]The wavelets were synthesized using the iterated filter bank algorithm, and the integrals involved in (6.11) and (6.12) were realized using high-precision numerical integration.

6.5   Multi-dimensional extension

We now extend the amplitude-phase representation to the multi-dimensional setting. Without the loss of generality, we derive the amplitude-phase representation for the particular two-dimensional setting.

We recall from Chapter 4 that the Gabor-like transform in two-dimension uses six analytic wavelets of the form

$$\Psi_l(\boldsymbol{x}) = \psi_l(\boldsymbol{x}) + j \, \mathcal{H}_{\theta_l} \psi_l(\boldsymbol{x})$$

where

$$\theta_1 = \theta_2 = 0, \quad \theta_3 = \theta_4 = \pi/2, \quad \theta_5 = \pi/4, \text{ and } \theta_6 = 3\pi/4.$$

The components of $\Psi_l(\boldsymbol{x})$ are oriented along the direction θ_l. The corresponding transform involves four biorthogonal systems. Let $\Psi_{l,i,\boldsymbol{k}}(\boldsymbol{x})$ be the dilated-translated copies of the dual wavelets $\tilde{\Psi}_1(\boldsymbol{x}), \ldots, \tilde{\Psi}_6(\boldsymbol{x})$, and let

$$c_{l,i}[\boldsymbol{k}] = \frac{1}{4} \int f(\boldsymbol{x}) \tilde{\Psi}_{l,i,\boldsymbol{k}}(\boldsymbol{x})^* \, d\boldsymbol{x}. \tag{6.13}$$

As before, we choose to reconstruct the signal by averaging the outputs . To derive the representation of $f(\boldsymbol{x})$ in terms of the coefficients $c_{l,i}[\boldsymbol{k}]$, we require the fractional extensions of the directional Hilbert transforms introduced in Chapter 2. Consider the shifted wavelets

$$\psi_l\big(\boldsymbol{x}; \tau_{l,i}[\boldsymbol{k}]\big) = \mathcal{H}_{\theta_l, \tau_i[\boldsymbol{k}]} \psi_l(\boldsymbol{x}).$$

where

$$\tau_{l,i}[\boldsymbol{k}] = \frac{1}{\pi} \arg(c_{l,i}[\boldsymbol{k}]). \tag{6.14}$$

As shown in Appendix A, using the invariances of fHTs, we can write

$$f(\boldsymbol{x}) = \sum_{l=1}^{6} \sum_{(i,\boldsymbol{k}) \in \mathbf{Z}^3} |c_{l,i}[\boldsymbol{k}]| \, \mathrm{Dil}_i \mathrm{Trans}_k \Big[\psi_l\big(\boldsymbol{x}; \tau_{l,i}[\boldsymbol{k}]\big) \Big]. \tag{6.15}$$

Thus, the signal is given by the superposition of six direction-selective wavelets with shifts.

As in the one-dimensional setting, further insight into the above representation is obtained by considering wavelets resembling windowed plane waves. In particular, let the wavelets be of the form

$$\psi_l(\boldsymbol{x}) = \varphi_l(\boldsymbol{x}) \cos\big(\Omega_l \boldsymbol{u}_{\theta_l}^T \boldsymbol{x}\big).$$

Then, we can write (6.15) as

$$f(x) = \sum_{l=1}^{6} \sum_{(i,k)\in\mathbf{Z}^3} \overbrace{\mathrm{Dil}_i \mathrm{Trans}_k[\varphi_l(x)]}^{\text{fixed window}} \mathrm{Dil}_i \mathrm{Trans}_k \left[\overbrace{|c_{l,i}[k]| \cos\left(\Omega_l u_{\theta_l}^T x + \pi \tau_{l,i}[k]\right)}^{\text{variable amp–phase plane wave}} \right]$$

(6.16)

This explicitly highlights the role of the phase-shifts in (6.14) as scale-dependent measures of the local signal displacements along the preferential directions.

This is the exactly scenario for the Gabor-like transforms discussed in Chapter 4. These are the asymptotic forms of the B-spline wavelet. We recall that (6.16) corresponds to the case where we analyze the signal using the dual complex wavelets, while the Gabor-like wavelets are used for reconstruction.

6.6 Shiftable spline wavelets

If $\psi(x)$ is not modulated, we can characterize (6.6) and (6.15) by studying the wavelets $\psi(x; \tau)$ and $\psi_l(x; \tau)$. The remarkable fact is that it can be done explicitly for the B-spline wavelets introduced in Chapter 5. To do so, we need the following result (see Appendix B for a proof), which is a generalization of Theorem 1 from Chapter 5.

Proposition 25 (Shifted spline wavelets). *The fractional Hilbert transform of a spline wavelet is again a spline wavelet of same order, but with different shift:*

$$\mathscr{H}_\gamma \psi_\tau^\alpha(x) = \psi_{\tau-\gamma}^\alpha(x).$$

(6.17)

That is, the fHT acts only on the shift parameter of the spline wavelet while preserving its genus and order. Thus, for the transform involving the spline wavelet basis and its Hilbert pair, we have the signal representation

$$f(x) = \sum_{(i,k)\in\mathbf{Z}^2} \mathrm{Dil}_i \mathrm{Trans}_k \left[|c_i[k]| \, \psi_{\tau-\tau_i[k]}^\alpha(x) \right].$$

We now consider the separable spline wavelets introduced in Chapter 5. It turns out that, as in the 1D case, the action is purely determined by the perturbation of the shift parameter of the constituent spline functions. However, the key difference is that the directional fHT operators act "differentially" on the shifts of the spline functions along each dimension. Observe that the six wavelets are of the general form

$$\sum_l g_l(x; \alpha, \tau_1) h_l(y; \alpha, \tau_2)$$

where $g_l(x; \alpha, \tau_1)$ and $h_l(y; \alpha, \tau_2)$ are spline scaling functions or wavelets which are of the same degree α, but whose shifts are τ_1 and τ_2. By introducing the shift

vector $\tau = (\tau_1, \tau_2)$, we can conveniently write this as $\psi_l(x; \alpha, \tau)$. For instance, the wavelets $\psi_1(x; \alpha, \tau)$ and $\psi_5(x; \alpha, \tau)$ are specified by (see Chapter 4)

$$\psi_1(x; \alpha, \tau) = \psi_{\tau_1}^\alpha(x) \beta_{\tau_2}^\alpha(y)$$

and

$$\psi_5(x; \alpha, \tau) = \frac{1}{\sqrt{2}} \left[\psi_{\tau_1}^\alpha(x) \psi_{\tau_2}^\alpha(y) - \psi_{\tau_1+1/2}^\alpha(x) \psi_{\tau_2+1/2}^\alpha(y) \right]$$

where we set $\tau_1 = \tau_2 = \tau$. In Appendix C we derive the following result.

Proposition 26. *Let $\tau = (\tau, \tau)$, and*

$$\mu_l = \begin{cases} 1 & \text{for } 1 \leq l \leq 4 \\ 1/\sqrt{2} & \text{for } l = 5 \text{ and } 6. \end{cases}$$

Then

$$\mathcal{H}_{\theta_l, \gamma} \psi_l(x; \alpha, \tau) = \psi_l \left(x; \alpha, \tau - \gamma \mu_l u_{\theta_l} \right).$$

The result is quite intuitive. The horizontal and vertical wavelets can be shifted along the direction of the corresponding directional fHT by perturbing the shift of the spline functions running along the same direction of the fHT; the shift of the spline function along the orthogonal direction remains unaffected. On the other hand, the diagonal wavelets can be shifted by perturbing the shift of the wavelets along either dimension.

As a direct consequence of (6.15) and Proposition 26, we have the non-asymptotic representation

$$f(x) = \sum_{l=1}^{6} \sum_{(i,k) \in \mathbf{Z}^3} \mathrm{Dil}_i \mathrm{Trans}_k \left[|c_{l,i}[k]| \, \psi_l \left(x; \alpha, \tau - \mu_k \tau_{l,i}[k] u_{\theta_l} \right) \right].$$

As discussed in §6.5, for sufficiently large α, $\psi_l(x; \alpha, \tau)$ constructed using the B-spline (orthonormal spline) wavelets resemble the Gabor (Shannon) wavelet where the shift $\tau_{l,i}[k]$ gets incorporated into the phase of the modulating plane wave.

Appendix A

Consider the four wavelet bases generated by the separable wavelets

$$\psi_{1+p}(x), \psi_{2+p}(x), \psi_{3+p}(x)$$

for $p = 0, 3, 6$ and 9. For a fixed p, we can represent $f(x)$ by the sereis

$$\sum_{(i,k) \in \mathbf{Z}^3} \left[a_{1+p,i}[k] \psi_{1+p,i,k}(x) + a_{2+p,i}[k] \psi_{2+p,i,k}(x) + a_{3+p,i}[k] \psi_{3+p,i,k}(x) \right]$$

where

$$a_{p+j,i}[k] = \int f(x)\tilde{\psi}_{p+j,i,k}(x)\,dx \qquad (j = 0, 1 \text{ and } 2). \tag{6.18}$$

We combine and regroup these into

$$
\begin{aligned}
f(x) = \sum_{(i,k)\in\mathbb{Z}^3} \Bigg[&\frac{1}{4}\Big(a_{2,i}[k]\psi_{2,i,k}(x) + a_{8,i}[k]\psi_{8,i,k}(x)\Big) \\
&+ \frac{1}{4}\Big(a_{5,i}[k]\psi_{5,i,k}(x) + a_{11,i}[k]\psi_{11,i,k}(x)\Big) \\
&+ \frac{1}{4}\Big(a_{1,i}[k]\psi_{1,i,[k]}(x) + a_{4,i}[k]\psi_{4,i,k}(x)\Big) \\
&+ \frac{1}{4}\Big(a_{7,i}[k]\psi_{7,i,k}(x) + a_{10,i}[k]\psi_{10,i,k}(x)\Big) \\
&+ \frac{1}{4\sqrt{2}}(a_{3,i}[k] - a_{12,i}[k])\left(\frac{\psi_{3,i,k}(x) - \psi_{12,i,k}(x)}{\sqrt{2}}\right) \\
&+ \frac{1}{4\sqrt{2}}(a_{6,i}[k] + a_{9,i}[k])\left(\frac{\psi_{6,i,k}(x) + \psi_{9,i,k}(x)}{\sqrt{2}}\right) \\
&+ \frac{1}{4\sqrt{2}}(a_{3,i}[k] + a_{12,i}[k])\left(\frac{\psi_{3,i,k}(x) + \psi_{12,i,k}(x)}{\sqrt{2}}\right) \\
&+ \frac{1}{4\sqrt{2}}(a_{6,i}[k] - a_{9,i}[k])\left(\frac{\psi_{6,i,k}(x) - \psi_{9,i,k}(x)}{\sqrt{2}}\right) \Bigg].
\end{aligned}
$$

The idea is to express the right-hand terms in terms of the coefficients

$$c_{l,i}[k] = |c_{l,i}[k]| \exp(j\pi\tau_{l,i}[k])$$

and the wavelets $\psi_{l,i,k}(x)$ and their Hilbert transforms. For example, consider the terms in the fifth and sixth line. From (6.13) and (6.18), we have

$$\frac{a_{3,i}[k] - a_{12,i}[k]}{4\sqrt{2}} = \frac{1}{4}\mathbf{Re}\int f(x)\tilde{\Psi}_{5,i,k}(x)\,dx = |c_{5,i}[k]|\cos(\pi\tau_{5,i}[k])$$

and

$$\frac{a_{6,i}[k] + a_{9,i}[k]}{4\sqrt{2}} = \frac{1}{4}\mathbf{Im}\int f(x)\tilde{\Psi}_{5,i,k}(x)\,dx = -|c_{5,i}[k]|\sin(\pi\tau_{5,i}[k]).$$

As for the wavelets, note that (cf. Chapter 4)

$$\psi_{5,i,k}(x) = \frac{\psi_{3,i,k}(x) - \psi_{12,i,k}(x)}{\sqrt{2}}$$

and

$$\mathcal{H}_{\theta_5} \psi_{5,i,k}(x) = \frac{\psi_{6,i,k}(x) + \psi_{9,i,k}(x)}{\sqrt{2}}.$$

By combining these, we get

$$|c_{5,i}[k]| \left(\cos \phi_i^5[k] \psi_{5,i,k}(x) - \sin \phi_i^5[k] \mathcal{H}_{\theta_5} \psi_{5,i,k}(x) \right) = |c_{5,i}[k]| \, \psi_{5,i,k}(x; \tau_i^5[k]).$$

Treating the remaining the terms identically, we have (6.15).

Appendix B

To establish (6.17), we consider the operator

$$\widehat{\Delta_\tau f}(\omega) = D_\tau(e^{j\omega})\hat{f}(\omega) \tag{6.19}$$

where

$$D_\tau(e^{j\omega}) = \begin{cases} (1 - e^{-j\omega})^\tau (1 - e^{j\omega})^{-\tau} & \text{for } \omega \in (-\pi, \pi)\backslash\{0\} \\ 0 & \text{when } \omega = 0. \end{cases}$$

Let $d_\tau[k]$ be the filter with Fourier transform $D_\tau(e^{j\omega})$. We claim that

$$\mathcal{H}_\gamma \beta_\tau^\alpha(x) = \Delta_\gamma \beta_{\tau-\gamma}^\alpha(x) = \sum_{k \in \mathbf{Z}} d_\gamma[k] \beta_{\tau-\gamma}^\alpha(x - k). \tag{6.20}$$

To see this, recall that

$$\widehat{\mathcal{H}_\gamma f}(\omega) = \exp\left[-j\pi\gamma \operatorname{sign}(\omega)\right]\hat{f}(\omega)$$

and note that

$$\hat{\beta}_\tau^\alpha(\omega) = \frac{D_\gamma(e^{j\omega})\hat{\beta}_{\tau-\gamma}^\alpha(\omega)}{(j\omega)^{-\gamma}(-j\omega)^\gamma}.$$

Then (6.20) is immediate once we notice that

$$(j\omega)^{-\gamma}(-j\omega)^\gamma = \exp\left[-j\pi\tau \operatorname{sign}(\omega)\right].$$

To proceed further, we need the identity

$$g_{\tau-\gamma}^\alpha[k] = d_\gamma * g_\tau^\alpha[k]. \tag{6.21}$$

A simple manipulation shows that

$$H_{\tau-\gamma}^\alpha(e^{j\omega}) = D_\gamma(-e^{-j\omega})H_\tau^\alpha(e^{j\omega}).$$

It follows that

$$
\begin{aligned}
G^{\alpha}_{\tau-\gamma}(e^{j\omega}) &= e^{j\omega}Q^{\alpha}(-e^{-j\omega})H^{\alpha}_{\tau-\gamma}(-e^{-j\omega}) \\
&= D_{\gamma}(e^{j\omega})\left[e^{j\omega}Q^{\alpha}(-e^{-j\omega})H^{\alpha}_{\tau}(-e^{-j\omega})\right] \\
&= D_{\gamma}(e^{j\omega})G^{\alpha}_{\tau-\gamma}(e^{j\omega})
\end{aligned}
$$

which establishes (6.21). From (6.20) and (6.21), we conclude that

$$
\begin{aligned}
\mathcal{H}_{\gamma}\psi^{\alpha}_{\tau}\left(\frac{x}{2}\right) &= \sum_{k\in\mathbf{Z}} g^{\alpha}_{\tau}[k]\mathcal{H}_{\gamma}\beta^{\alpha}_{\tau}(x-k) \\
&= \sum_{k\in\mathbf{Z}} g^{\alpha}_{\tau}[k]\left\{\sum_{n\in\mathbf{Z}} d_{\gamma}[n]\beta^{\alpha}_{\tau-\gamma}(x-k-n)\right\} \\
&= \sum_{m\in\mathbf{Z}} g^{\alpha}_{\tau}*d_{\gamma}[m]\beta_{\tau-\gamma}(x-m) \\
&= \psi^{\alpha}_{\tau-\gamma}\left(\frac{x}{2}\right).
\end{aligned}
$$

Appendix C

We derive the relation for the wavelets $\psi_1(x;\alpha,\tau)$ and $\psi_5(x;\alpha,\tau)$. The rest can be derived identically.

Form (6.17), one of the relations is immediate,

$$
\mathcal{H}_{0,\gamma}\psi_1(x;\alpha,\tau) = \mathcal{H}_{\gamma}[\psi^{\alpha}_{\tau}(x)]\beta^{\alpha}_{\tau}(y) = \psi^{\alpha}_{\tau-\gamma}(x)\beta^{\alpha}_{\tau}(y) = \psi_1(x;\alpha,\tau-\gamma u_{\theta_1}).
$$

For the second relation, we require the factorization

$$
\mathcal{H}_{\frac{\pi}{4},\tau} = \mathcal{H}_{0,\frac{\tau}{2}}\mathcal{H}_{\frac{\pi}{2},\frac{\tau}{2}}.
$$

It can be verified that this is true for functions whose frequency supports are restricted to the quadrants $\{(\omega_1,\omega_2):\omega_1>0,\omega_2>0\}$ and $\{(\omega_1,\omega_2):\omega_1<0,\omega_2<0\}$. Since this is true for $\psi_5(\omega;\alpha,\tau)$, by (6.17), we have

$$
\begin{aligned}
\mathcal{H}_{\frac{\pi}{4},\gamma}\psi_5(x;\alpha,\tau) &= \frac{1}{\sqrt{2}}\mathcal{H}_{0,\frac{\gamma}{2}}\mathcal{H}_{\frac{\pi}{2},\frac{\gamma}{2}}\left[\psi^{\alpha}_{\tau}(x)\psi^{\alpha}_{\tau}(y)-\psi^{\alpha}_{\tau+\frac{1}{2}}(x)\psi^{\alpha}_{\tau+\frac{1}{2}}(y)\right] \\
&= \frac{1}{\sqrt{2}}\left[\psi^{\alpha}_{\tau-\frac{\gamma}{2}}(x)\psi^{\alpha}_{\tau-\frac{\gamma}{2}}(y)-\psi^{\alpha}_{\tau-\frac{\gamma}{2}+\frac{1}{2}}(x)\psi^{\alpha}_{\tau-\frac{\gamma}{2}+\frac{1}{2}}(y)\right] \\
&= \psi_5\left(x;\alpha,\tau-\gamma\frac{1}{\sqrt{2}}u_{\theta_5}\right).
\end{aligned}
$$

Chapter 7

Stereo matching using a translation-invariant wavelet pyramid

Abstract — State-of-the-art algorithms for dense stereo matching perform a global optimization on graphs constructed from the raw pixel intensities. The most popular ones are based on Graph-Cuts, Belief Propagation, and Dynamic Programming. Graph-based algorithms, however, tend to be rather slow when both the size of the stereo images and the range of disparities are large [61].

Image pyramiding provides an attractive means of accelerating such algorithms. In this chapter, we propose a coarse-to-fine stereo-matching algorithm that does narrow-band DP on the Gabor-like wavelet pyramid introduced in Chapter 5. The crucial feature of our pyramid is that it provides near translation-invariance at the cost of moderate redundancy. We show that a significant reduction of the computation time is obtained in comparison to the standard DP algorithm.

We evaluate our algorithm on the benchmark Middlebury database maintained by Scharstein and Szeliski [62].

7.1   Introduction

The goal of a stereo matching algorithm is to establish pixel correspondences between a pair of stereo images. For the present work, we assume that the images are *rectified*, that is, the epipolar lines are aligned with the corresponding scanlines [63]. In this case, the correspondence is simply given by the local displacement of the pixel along the scanline. The output of a stereo algorithm is a 2-dimensional

map that records the disparity for every pixel in one image (the reference image). We call this the *disparity map*.

Let us denote the left and right image by $I_1[m,n]$ and $I_2[m,n]$. The rows are indexed by $1 \le m \le M$, where M is the height of the image, while the columns are indexed by $1 \le n \le N$, where N is the width of the image.

The advantage with rectified images is that one can estimate the disparity on a scanline-by-scanline basis. In this case, the stereo matching problem is reduced to one of finding correspondences between the 1-dimensional profiles $I_1[m,\cdot]$ and $I_2[m,\cdot]$ for every $1 \le m \le M$. Ideally, the goal is one of finding a disparity map $(d_{m,n})$ such that

$$I_2[m, n + d_{m,n}] = I_1[m,n]. \tag{7.1}$$

In practice, one usually proceeds by posing this as an optimization problem [63].

7.2   Stereo-matching using optimization

We begin by setting up the optimization strategy used for estimating the disparity. To do so, we use the terminology of Scharstein et al. [61].

Let the basic cost of matching the left pixel $I_1[m,n]$ to the right pixel $I_2[m,n']$ be

$$|I_2[m,n'] - I_1[m,n]|. \tag{7.2}$$

We define the *aggregated cost* by averaging the basic cost over a fixed neighborhood:

$$\varrho\big(I_1[m,n], I_2[m,n']\big) = \sum_{|p| \le \Omega_p} \sum_{|q| \le \Omega_q} |I_2[m-p, n'-q] - I_1[m-p, n-q]|.$$

Note that we also average the cost from the adjacent scanlines. This introduces some consistency between adjacent scanlines, and helps reduce the artifact of "horizontal streaking" that is typical for scanline-based optimization [64, 61].

In the so-called *local optimization* method, one estimates the disparity by minimizing the aggregated cost pixel-by-pixel:

$$d_{m,n} = \arg \min_{0 \le d < \Delta} \varrho\big(I_1[m,n], I_2[m, n+d]\big) \tag{7.3}$$

where Δ is an a priori bound on the range of disparity. This is popularly known as the *Winner-Take-All* strategy in computer vision [61]. The main difficulty here is that the disparity at every pixel is computed independently of its neighboring pixels. As a result, the final disparity map tends to be noisy.

We can solve this problem using regularization; that is, by introducing a smoothness constraint on the disparity map. This leads us to the *global optimization* methods, where we allow the pixels to be assigned disparities which are possibly sub-optimal in terms of (7.3). The idea is to offset this extra cost by increasing the agreement of the pixel with its neighbors in terms of their disparities.

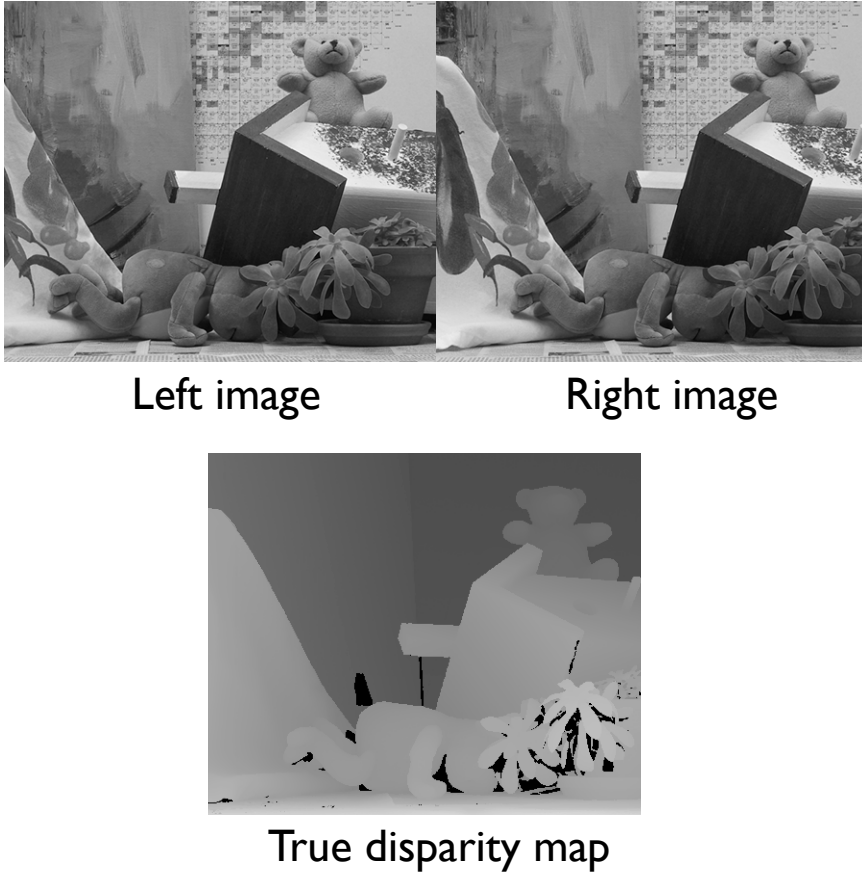Left image   Right image

True disparity map

Figure 7.1: A particular example of stereo images. This shows the rectified *Teddy* stereo pair and the corresponding disparity map. The images were obtained from the Middlebury database [62]. The image size is 450 × 375, and the maximum disparity is 55 pixels.
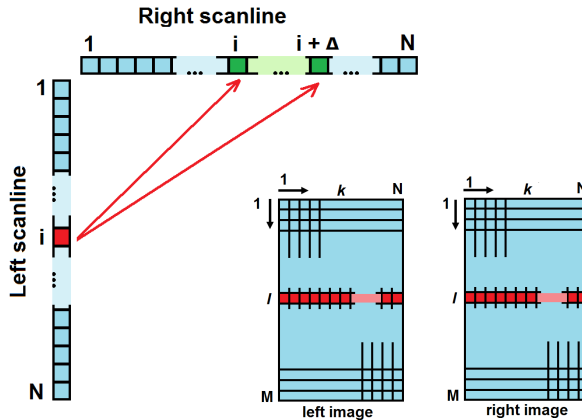
Figure 7.2: Scanline-based stereo matching. In this case, the left and right images are assumed to be rectified, and one proceeds to estimate the disparity on a scanline-by-scanline basis in which every pairs of rows from the left and right image are matched independently. This turns the 2-dimensional stereo-matching problem into a 1-dimensional problem resulting in a significant reduction of the computational complexity.

We begin by selecting a particular row. Let $d_1, \cdots, d_N$ be the disparity along this row. For notational convenience, we drop the suffix corresponding to the row. The global cost function we want to minimize is given by

$$C(d_1, \cdots, d_N) = C_{\text{data}}(d_1, \cdots, d_N) + \lambda C_{\text{smooth}}(d_1, \cdots, d_N) \tag{7.4}$$

where

$$C_{\text{data}}(d_1, \cdots, d_N) = \sum_{n=1}^{N} \varrho(I_1[m, n], I_2[m, n + d_n]) \tag{7.5}$$

and

$$C_{\text{smooth}}(d_1, \cdots, d_N) = \sum_{n=1}^{N-1} |d_n - d_{n+1}|. \tag{7.6}$$

The role of the data term (7.5) is to penalize the disagreement of the disparity map with the observed data. The smoothness term (7.6) is the total-variation norm of the disparity profile. It penalizes profiles which are not piecewise-smooth. This particular regularization is known to perform well for stereo matching [65].

7.2.1   Fast computation of aggregated cost

In the sequel, we will be required to compute the aggregated cost in (7.5) at different positions in the image and for disparities in the range $\{0, 1, \ldots, \Delta - 1\}$. It turns out

that this can be efficiently done by introducing a 3-dimensional matrix $\mathbf{V}(m,n,d)$ of size $M \times N \times \Delta$. We call this the *cost volume*. The elements of the cost volume are set using (7.2),

$$\mathbf{V}(m,n,d) = |I_1[m,n] - I_2[m,n+d]|.$$

Then

$$\varrho\left(I_1[m,n], I_2[m,n']\right) = \sum_{|p| \leq \Omega_p} \sum_{|q| \leq \Omega_q} \mathbf{V}(m-p, n-q, n'-n).$$

This suggests the following efficient algorithm for computing the aggregated cost. For every fixed d, we convolve the 2-dimensional profile $\mathbf{V}(\cdot, \cdot, d)$ with the box distribution

$$\text{Box}[m,n] = \begin{cases} 1 & \text{if } |m| < \Omega_p \text{ and } |n| < \Omega_q \\ 0 & \text{otherwise.} \end{cases}$$

As is well-known, this can be computed with $O(N \times M)$ operations using running-sums and finite differences, irrespective of the size of Ω_p and Ω_q [28]. This is particularly efficient when the size of the window is large. Let $\mathbf{V}_*(m,n,d)$ be the resulting matrix. Then

$$\varrho\left(I_1[m,n], I_2[m,n']\right) = \mathbf{V}_*(m,n,n'-n).$$

7.2.2 Dynamic programming

The naive optimization of (7.4) using exhaustive search has the complexity $O(N^\Delta)$. One can do better by observing that the problem of finding correspondences between scanlines can be cast as a *string matching* problem, along with some additional ordering constraints. As is well-known, the string matching problem can be efficiently solved using dynamic programming [66]. Dynamic programming (DP) proceeds by breaking the complex problem into simpler subproblems, solves these subproblems, and finally combines them to obtain the global solution. In particular, we can optimize (7.4) in $O(N\Delta)$ using DP. Dynamic programming was first used for edge-based approaches, e.g., by Ohta and Kanade in [64]. Geiger et al. were among the first to apply DP for stereo matching based on pixel-wise intensity differences [67].

We set up the framework for minimizing (7.4) using DP as follows. Our approach is motivated by the algorithm in [68]. We first fix a row m in the reference image. We then initialize a matrix $\mathbf{M}$ of size $N \times \Delta$ and copy the plane of $\mathbf{V}_*$ corresponding to row m into $\mathbf{M}$:

$$\mathbf{M}(n,d) = \mathbf{V}_*(m,n,d). \tag{7.7}$$

We call $\mathbf{M}$ the *score matrix* (cf. Figure 7.3). We next proceed to update the entries of
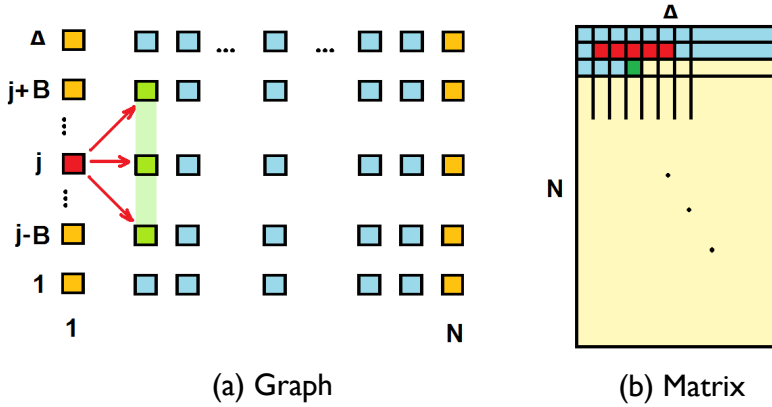
(a) Graph   (b) Matrix

Figure 7.3: The data abstractions used for matching the respective left and right scanlines using dynamic programming. (a) The ordered graph structure where every node corresponds to a given position in the scanline (order from left to right) and is assigned a fixed disparity (ordered from top to bottom). The edges between adjacent columns represent the disparity transition between a pixel and its neighbor. (b) The score matrix is of size $N \times \Delta$. This is used to assign different scores to different nodes in the graph during the first phase of dynamic programming. The matrix is initialized using (7.7). It is then updated from top to bottom using the rule in (7.8).

M in a top-to-bottom fashion starting from the first row,

$$\mathbf{M}(n,d) \leftarrow \mathbf{M}(n,d) + \min \Big\{ \mathbf{M}(n-1, d-B) + \lambda B, \cdots$$
$$\cdots, \mathbf{M}(n-1, d), \cdots, \mathbf{M}(n-1, d+B) + \lambda B \Big\}. \qquad (7.8)$$

The number B is the maximum allowed "jump" (or transition) in disparity that we allow between two adjacent pixels. Following the smoothness constraint in (7.6), every jump incurs a cost λ. We store the index of the pixel that gives the minimum in (7.8).

After the last row $\mathbf{M}(N, \cdot)$ is filled, we get the optimal cost simply by taking the minimum over this row. The optimal path is then obtained by back-tracking on the indices stored at the time of filling the score matrix. The optimal path gives us the disparity at each pixel in the scanline. The above process is repeated over all the scanlines to generate the full disparity map.
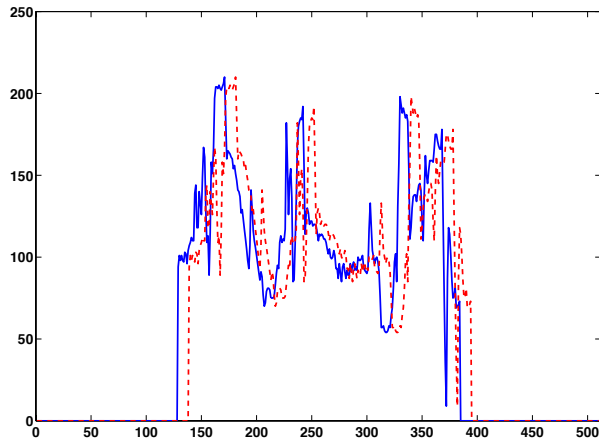
Figure 7.4: The reference scanline $f(x)$, shown in BLUE (solid line), and its translated $f(x - h)$, shown in RED (dashed line). We have set $h = 10$ pixels. The wavelet decompositions of these scanlines using the standard wavelet pyramid and our complex wavelet pyramid are compared in Figure 7.5.

7.3   Use of multiresolution pyramids

Multiresolution techniques, particularly the ones based on wavelet pyramid, provide an attractive framework for the estimation of stereo correspondence [69, 70]. The main advantage with these representations is their hierarchical nature, which (among other things) allows one to reduce the dimensionality of the problem. From a computational standpoint, this allows the usage of the so-called *coarse-to-fine* algorithms, where results from coarser levels are used to constrain the search at finer levels. This class of algorithms may be considered as being intermediate to those where the correspondence is established using either local search [71, 72], or by optimizing a global cost function [73].

There are also some added advantages with wavelet based matching. Conventional approaches based on intensity-based matching estimate the disparity from the raw pixel intensities directly. This makes them susceptible to level shifts; e.g., due to illumination variation between the left and right camera. By using a wavelet representation, one can easily cancel out such level shifts (bias invariance). Multiresolution strategies have also proven useful in resolving the ambiguities of dense matching of smooth surfaces which lack clear feature points and have few occlusions [74].

7.4   Translation-invariant wavelet pyramid

In the context of stereo matching, the common problem with real wavelet pyramids is their poor translation-invariance [23, 69]. Complex wavelet pyramids provide an attractive means of improving the translation-invariance at the cost of moderate redundancy, that is, using two pyramids instead of one. For example, Magarey et al. [75, 76] have introduced algorithms for motion estimation and image matching using complex Gabor filters.

For our stereo-matching algorithm, we use the complex wavelet pyramid based on the complex Gabor-like wavelets introduced in Chapter 5. We recall that the analysis wavelet in this case is (approximately) given by

$$\Psi(x) = \varphi(x) \exp(j\omega_0 x) \tag{7.9}$$

where $\varphi(x)$ is a Gaussian window, and ω_0 is the modulation frequency.

It turns out that the amplitude of this complex wavelet pyramid exhibits near translation-invariance (modulo the decimation). It provides a satisfactory encoding of the local image displacements in the amplitude function. To explain this, we consider a scanline $f(x)$ and its uniform translate $f(x - h)$. The complex wavelet coefficient of at scale 2^{-i} and translation $2^i k$ is given by

$$c_i[k] = 2^{-i/2} \int f(x) \Psi^*(2^{-i} x - k) \, dx.$$

On the other hand, the wavelet coefficient of the translated scanline is given by

$$\tilde{c}_i[k] = 2^{-i/2} \int f(x - h) \Psi^*(2^{-i} x - k) \, dx. \tag{7.10}$$

It is shown in Appendix A that

$$\tilde{c}_i[k] = c_i[k - n] \exp(-j\omega_0 2^{-i} \Delta h) + \text{ residual}$$

where the size of the "residual" is within $2^{-i} |\Delta h| \, \|\varphi'\|_\infty \, |\hat{f}(2^{-i}\omega_0)|$, and where $2^i n$ is the integer approximation of h and Δh is the fractional component:

$$h = 2^i n + \Delta h. \tag{7.11}$$

Note that the integer component of h gets encoded in the translation of the amplitude function, while the fractional component Δh gets encoded in the phase function.

Clearly, the residual is zero[1] when $\Delta h = 0$. In general, since $\|\varphi'\|_\infty$ is small for the Gaussian, we have the close approximation

$$\tilde{c}_i[k] \approx c_i[k - n] \exp(-j\omega_0 2^{-i} \Delta h).$$

---

[1]This is also the case if $f(x)$ is bandlimited to $[-2^{-J}\omega_0, 2^{-J}\omega_0]$, where 2^{-J} is the coarsest scale.
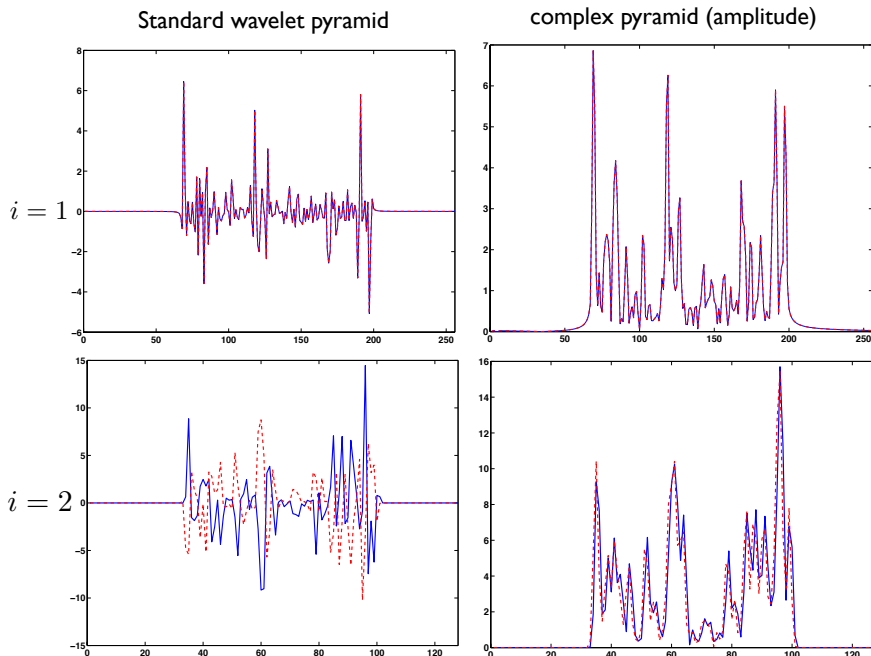
Figure 7.5: Response to uniform translation. **Left**: The coefficients of the translated signals shown in Figure 7.4 for the standard wavelet pyramid. **Right**: The amplitude functions $|\tilde{c}_i[k]|$ and $|c_i[k-n]|$ of the translated signals of the Gabor-like wavelet pyramid. **Level** $i = 1$: The amplitude functions at scale $i = 1$, which match perfectly. In this case, $n = 5$ and $\Delta h = 0$. **Level** $i = 2$: The amplitude functions at scale $i = 2$; $n = 2$ and $\Delta h = 2$.

Taking the modulus, we see that the amplitude is approximately translation-invariant (modulo the decimation):

$$|\tilde{c}_i[k]| \approx |c_i[k-n]| \qquad (k \in \mathbf{Z}). \tag{7.12}$$

We note that it is rather difficult to arrive at a comparable approximation using real wavelets. The point is that the Gabor wavelet $\Psi(x)$ is given by the product of a smooth window $\varphi(x)$ and a separate oscillatory component $\exp(j\omega_0 x)$. As a result, we can control the error generated by the fractional translation Δh by making the window sufficiently smooth. On the other hand, using the addition-multiplication property of complex exponentials and by taking the modulus, we are able to cancel out the phase component. As shown in Appendix A, in case of real wavelets,

$$\tilde{c}_i[k] = c_i[k-n] + \text{residual}$$

where the residual is within $2^{-i}|\Delta h|\,\|\psi'\|_\infty\,\|f\|_1$. Now $\|\psi'\|_\infty$ is typically high for a
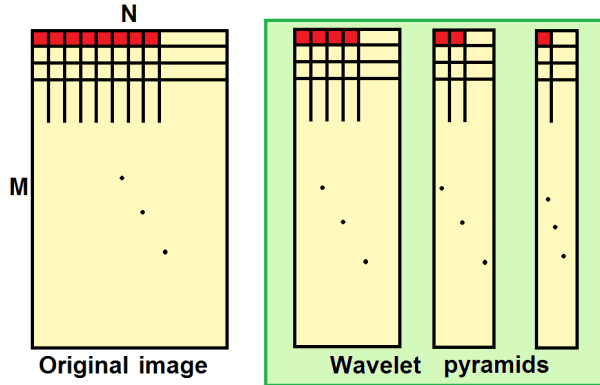
Figure 7.6: Original image and its wavelet pyramid. This shows a three level decomposition of the raw pixel intensities of every scanline of the original image. The resolution of the original image is $M \times N$, where N is the width of the scanline. The pyramids are of resolution (left to right) $M \times 2^{-1}N, M \times 2^{-2}N$, and $M \times 2^{-3}N$. We use a wavelet pyramid for each of the left and right images. The disparity estimation begins at the coarsest resolution and progresses towards the finer resolutions.

real wavelet owing to its oscillatory nature. Also, $\|f\|_1 \geq |\hat{f}(\omega)|$. By comparing the residuals in the two cases, we now see why it is rather difficult for the real wavelet to achieve the quality of translation-invariance as achieved by the complex counterpart. We provide an empirical justification of these above observations using an example in Figure 7.5.

7.5   Stereo-matching algorithm

Our objective is to accelerate the standard DP algorithm using the Gabor wavelet pyramid. The idea is to estimate the disparity by matching the amplitudes of the left and right pyramids. As mentioned earlier, the advantage with the wavelet pyramid is that it offers dimensionality reduction. At every sub-resolution, the size of the (left and right) images and the range of the disparity is reduced by a factor of two. This allows us to cuts down the complexity of the DP by at least a factor of four.

The down side, however, is that we loose the local "fractional" displacements. For example, in the case of uniform translation, we loose the part Δh in (7.11). It is only the part n in (7.11) that can be captured using the left and right wavelet pyramids. At scale 2^i, the best estimate that we can have is $h \approx 2^i n$. Our idea is to recover the lost Δh using a *narrow-band* DP at the finer scale of the pyramid (cf. Figure 7.7).
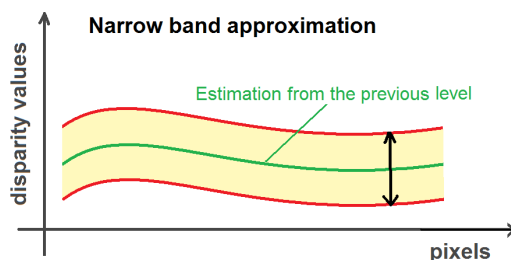
Figure 7.7: Narrowband update of the disparity. The disparity from the coarser resolution is used to initialize the dynamic programming for the finer resolution. At the coarsest resolution, we use a full search for estimating the disparity. At every subsequent finer resolution, we restrict the search to a "narrow" neighborhood of the estimate obtained from the coarser resolution. This allows us to correct the disparity information that is lost during the downsampling process.

7.5.1 Gabor wavelet pyramid

We use two Gabor wavelet pyramids corresponding to the images $I_1[m,n]$ and $I_2[m,n]$. We perform a J-level decomposition of every scanline using the Gabor-like transform described in Chapter 5. At every level $1 \leq i \leq J$, we take the modulus of the wavelet coefficients of each scanline, and concatenate them into a single image (cf. Figure 7.6). This gives us the left and right pyramids

$$A_1[i,m,n] \quad \text{and} \quad A_2[i,m,n] \quad (1 \leq i \leq J).$$

The images $A_1[i,m,n]$ and $A_2[i,m,n]$ are of size $M \times 2^{-i}N$. The idea is to match the profiles $A_1[i,m,\cdot]$ and $A_2[i,m,\cdot]$ for every $1 \leq m \leq M$ to estimate the disparity map at level i. We denote this map by $d_i[m,n]$.

7.5.2 Coarse-to-fine refinement

We first estimate the disparity at the coarsest level of the wavelet pyramid. We do this independently for each scanline by minimizing (7.4). We use the standard DP described in §7.2.2. The main modifications that we make are as follows.

(1) We act on the pyramids instead of raw pixel intensities. That is, we set

$$I_1[m,n] = A_1[J,m,n] \quad \text{and} \quad I_2 = [m,n] = A_2[J,m,n]$$

in (7.5).

(2) The unknown set of disparities in (7.4) is given by $d_1, \cdots, d_L$, where $L = 2^{-J}N$. The range of disparity goes down by the same factor, that is, $0 \leq d_l \leq 2^{-J}\Delta$ for every $1 \leq l \leq L$.

We run a DP for each of the M rows. This gives us the computed disparity map $d_J[m,n]$.

We then iteratively propagate this estimate to the next scale of the pyramid. The main point is that the disparity estimate obtained from the coarser level is used to "initialize" the DP for the next level. We call this the *narrow-band* DP. It consists of the following steps.

(1) Suppose we have the map $d_i[m,n]$ for some $1 \leq i \leq J$. If $i = 1$, we simply use the left and right images in (7.5). Otherwise, we set

$$I_1[m,n] = A_1[i-1,m,n] \quad \text{and} \quad I_2 = [m,n] = A_2[i-1,m,n].$$

(2) The unknown set of disparities in (7.4) is given by $d_1,\ldots,d_L$, where $L = 2^{-i+1}N$. These are obtained by extrapolating the coarse estimate $d_i[m,n]$ both in space and range, and then refining them within a narrow window. A straightforward way of doing this is by setting

$$d_l = \begin{cases} \delta_l + 2d_i[m,(l+1)/2] & \text{if } l \text{ is odd} \\ \delta_l + 2d_i[m,l/2] & \text{if } l \text{ is even,} \end{cases}$$

where

$$|\delta_l| \leq \eta \, 2^{-i+1}\Delta \qquad (1 \leq l \leq L).$$

The factor $\eta << 1$ is used to determine the size of narrow-band. Note that the optimization variables in this case are $\delta_1,\ldots,\delta_L$. The score matrix $\mathbf{M}$ in this case is of size $L \times \eta \, 2^{-i+2}\Delta$.

We run a DP for each of the M rows to get the disparity map $d_{i+1}[m,n]$. We keep doing this till we get to the image space.

7.6   Experiments

We compare the results and execution time of our algorithm with local optimization and the standard DP algorithm. These represent the two extremes as far as the execution speed and the quality of reconstruction is concerned. Local optimization is very fast, but the computed disparity tends to be very noisy. On the other hand, DP uses regularization to smooth out the noise and yields better results as a consequence. The flip side is that it runs slow on large images, particularly when the disparities are large.

To judge the performance of our algorithm, we use the three different test images shown in Figures 7.1, 7.9, and 7.11. The disparity estimates obtained using the three methods (along with the true disparity) are shown in Figures 7.8, 7.10, and 7.12. We have cropped the disparity maps along their boundaries. The parameter settings along with the execution times are also mentioned. These have been manually tuned to obtain the best result in every case. In all the three cases, we used a three level

(a) True disparity

(b) Local optimization

(c) Dynamic programming

(c) Our method
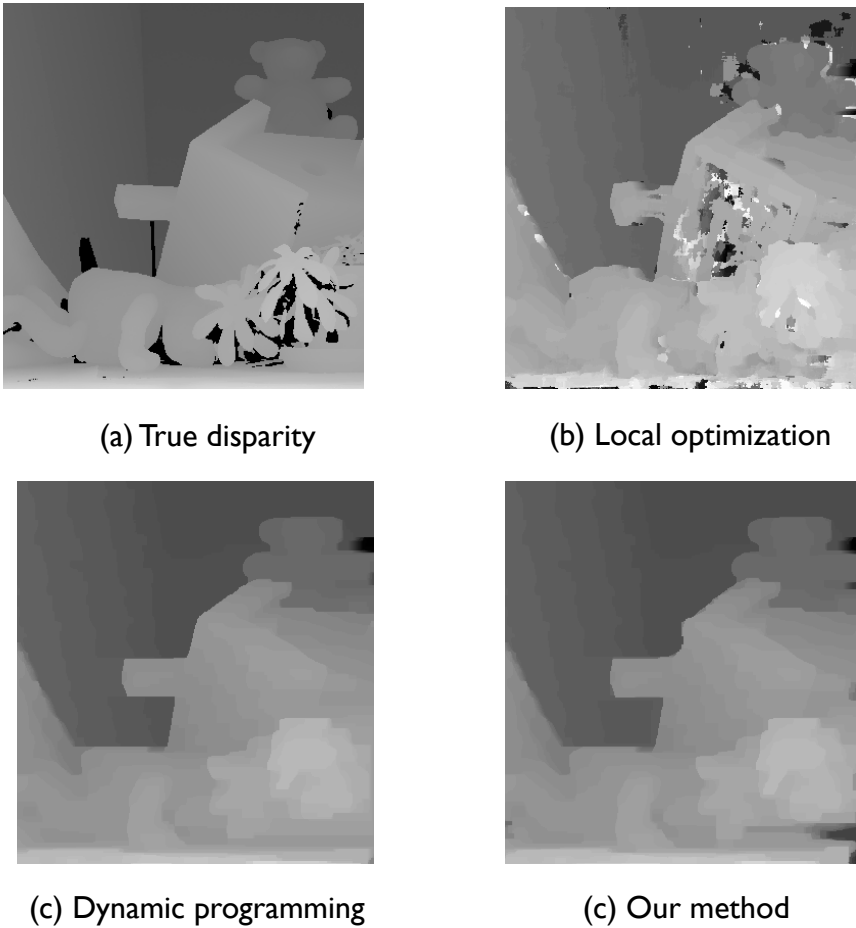
Figure 7.8: Comparison of the disparity maps obtained using local optimization, dynamic programming (DP), and our wavelet method (WDP) for the stereo pair in Figure 7.1. The parameters have been manually tuned to obtain the best results. As a result of the regularization, the disparities obtained from DP and WDP are seen to be less noisy than that obtained from local optimization. The results of DP and WDP are seen to be comparable. The execution time for WDP is marginally less then DP in this case. The parameters used for WDP are: $\Delta = 110, \Omega_p = 5, \Omega_q = 5, B = 37$, and $\lambda = 0.06$.

wavelet decomposition using the Gabor-like cubic B-spline wavelet. The images have been been padded along their width to their nearest dyadic size. The wavelet pyramids were built on these padded images. While estimating the disparity, we simply restricted the computation within the true boundaries.

As seen in Figure 7.8, the disparity map obtained from local optimization is very noisy for the pair in Figure 7.1. Thanks to the regularization, the outputs of DP and WDP are seen to be less noisy. The results of DP and WDP are seen to be comparable. Since the size of the image and the range of disparity is moderate, the execution time for WDP is marginally less then DP in this case. The parameter settings for WDP are: $\Omega_p = 5, \Omega_q = 5, B = 37$, and $\lambda = 0.06$.

In Figure 7.10, we show the disparity estimates obtained using different algorithms for the stereo pair in Figure 7.9. The results obtained using our method is seen to be comparable to the one using DP. However, we are able to obtain a significant reduction in the execution time using wavelet pyramiding and coarse-to-fine refinement. The execution time for DP is 46 seconds, while our method takes 16 seconds. The parameters used in our method are: $\Omega_p = 5, \Omega_q = 5, B = 18$, and $\lambda = 0.13$.

Finally, the results obtained for the images in Figure 7.11 are shown in Figure 7.12. The estimate obtained using DP is again seen to be better than that obtained using local optimization. Due to the large size of the image and the disparities, DP takes a total of 81 seconds for computing the disparity. Using wavelet pyramiding along with narrow-band DP, we are able to reduce this to 43 seconds. The parameters used in this case are: $\Omega_p = 5, \Omega_q = 5, B = 60$, and $\lambda = 0.06$.

7.7   Future extensions

We presented a preliminary version of a stereo matching algorithm based on a translation-invariant wavelet pyramid. We showed that the execution time for the DP can be cut down by $50 - 70\%$ using coarse-to-fine refinement, without any appreciable degradation of the disparity map. There is still a lot of scope of improving our algorithm on various grounds. These are as follows.

- **Occlusion handling**. We used a very basic DP algorithm which is unable to handle occlusions. This is because we forced a match between every left and right pixel. It is, however, possible to modify the DP algorithm so as to handle occlusions. This can be done, for example, by introducing the option of not matching certain pixels, and an appropriate penalty for doing so into the cost funcion.

- **Sub-pixel estimation using phase**. The disparity estimates returned using narrow-band DP were all integer-valued. As a result, the disparity map often exhibited contouring effects in regions where there are no smooth transitions between different disparities. This can be fixed by incorporating sub-pixel refinement. In our case, this can be done (at no extra cost) using the phase

of the complex wavelet coefficients. The phase information can be used to retrieve the "fractional" displacements which are not encoded in the amplitude functions.

- **Adaptive narrow-band DP**. During the narrow-band DP, we allowed for disparity refinements within a certain tolerance. This tolerance η was fixed globally at every scale. From our experiments, it was clear that pixels situated near the boundaries of disparity transitions required a higher tolerance than those situated in the interior of regions having uniform disparities. This suggests the possibility of adapting the tolerance η at every point via some means, e.g., using an estimate of the local variation of the disparity.

- **Use of 2-dimensional wavelet transform**. We could exploit the redundancy along the vertical direction by subsampling the rows as well. This could be done using the 2-dimensional Gabor-like wavelet transform proposed in Chapter 5. For example, we can use the subbands that are near-translation invariant along the horizontal and vertical directions. This would further reduce the computational time.

Appendix A

Using (7.9), we can write (7.10) as

$$\tilde{c}_i[k] = 2^{-i/2} \int f(x)\varphi(2^{-i}x - k + 2^{-i}h)\exp\left[-j\omega_0(2^{-i}x - k + 2^{-i}h)\right] dx.$$

For some fixed scale $i > 0$, we decompose the translation as

$$h = 2^i n + \Delta h$$

where $2^i n$ is the integer approximation of h at scale 2^i, and $\Delta h = h - 2^i n$. The former is the "large" integer component of the translation (relative to the scale 2^{-i}), while the latter is the "small" fractional component. Using this decomposition, we can write

$$\tilde{c}_i[k] = \exp(-j2^{-i}\omega_0\Delta h)$$
$$\left[2^{-i/2}\int f(x)\varphi(2^{-i}x - k + n + 2^{-i}\Delta h)\exp\left[-j\omega_0(2^{-i}x - k + n)\right] dx\right].$$

To estimate the drop in the height of the window $\varphi(x)$, we use the first-order approximation

$$\varphi(x + \tau) = \varphi(x) + \varepsilon(\tau)$$

where $|\varepsilon(\tau)| \leq \tau\|\varphi'\|_\infty$. A simple manipulation shows that

$$\tilde{c}_i[k] = c_i[k - n]\exp(-j\omega_0 2^{-i}\Delta h)$$
$$+ \varepsilon(2^{-i}\Delta h)\hat{f}(2^{-i}\omega_0)\exp\left[-j\omega_0(2^{-i}\Delta h - k + n)\right].$$

In particular,

$$\tilde{c}_i[k] = c_i[k-n]\exp(-j\omega_0 2^{-i}\Delta h) + \text{ residual}$$

where the size of the "residual" is within $2^{-i}|\Delta h|\,\|\varphi'\|_\infty\,|\hat{f}(2^{-i}\omega_0)|$.

On the other hand, an identical analysis for a real wavelet $\psi(x)$ shows that

$$\tilde{c}_i[k] = c_i[k-n]\exp(-j\omega_0 2^{-i}\Delta h) + \text{ residual}$$

where the size of the residual in this case is within $2^{-i}|\Delta h|\,\|\psi'\|_\infty\,\|f\|_1$.

Left image   Right image
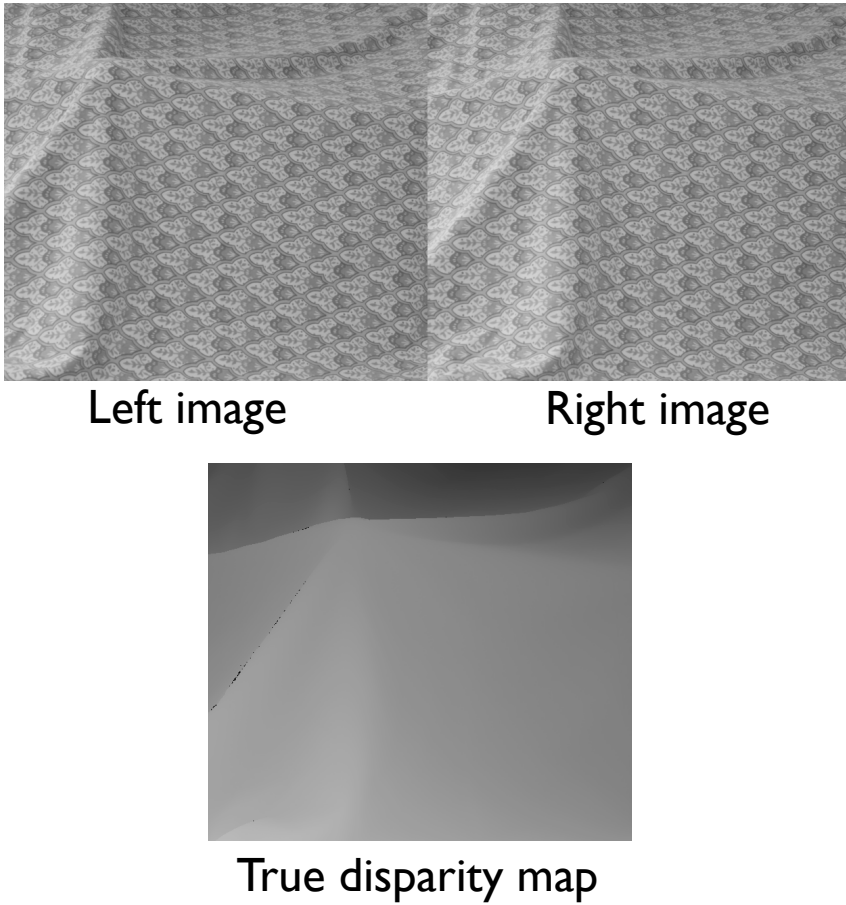
True disparity map

Figure 7.9: Rectified stereo images of *Cloth1* and the true disparity map [62]. The image size is 626 × 555, and the maximum disparity is 110 pixels.

(a) True disparity

(b) Local optimization

(c) Dynamic programming
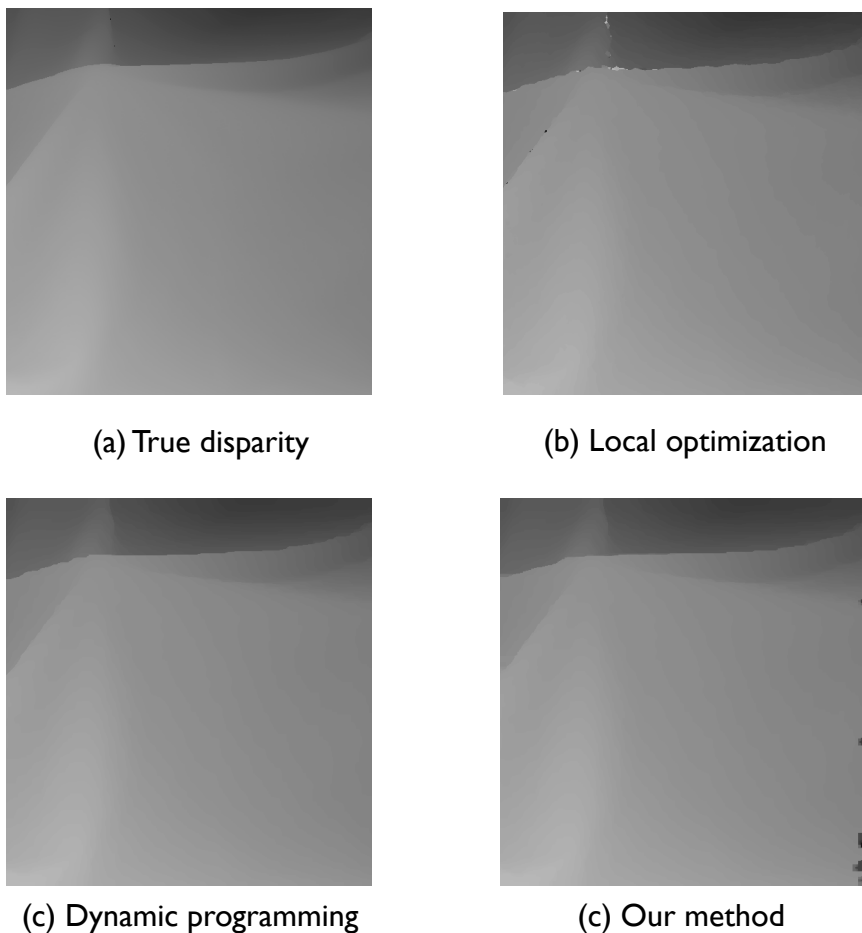
(c) Our method

Figure 7.10: Disparity estimates obtained using different algorithms for the stereo pair in Figure 7.9. The results obtained using our method is seen to be comparable to the one using DP. However, we are able to obtain a significant reduction in the execution time using wavelet pyramiding and coarse-to-fine refinement. The execution time for DP is 46 seconds, while our method takes 16 seconds. The parameters used in our method are: $\Omega_p = 5, \Omega_q = 5, B = 18$, and $\lambda = 0.13$.
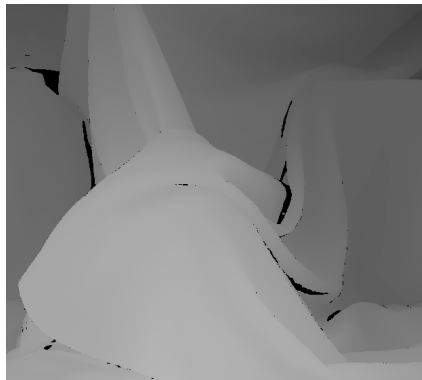
Figure 7.11: *Cloth2* stereo pair and the true disparity map [62]. The size of the image is 626 × 555, and the maximum disparity is 170 pixels.
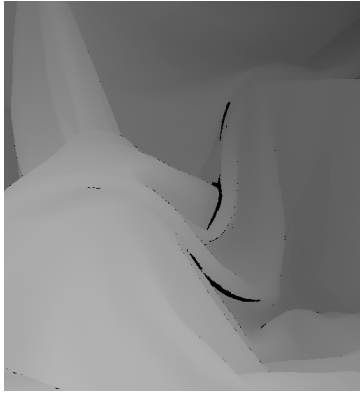
(a) True disparity
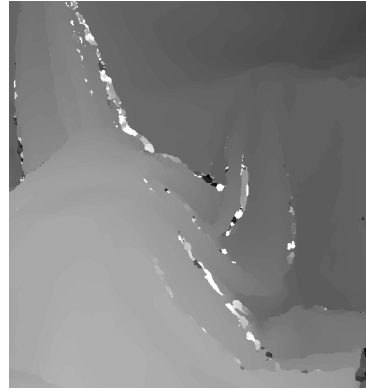
(b) Local optimization

(c) Dynamic programming
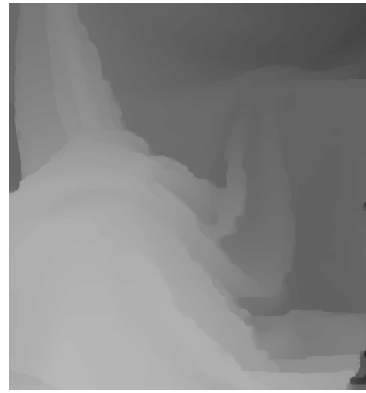
(c) Our method

Figure 7.12: Results obtained for the images in Figure 7.11. The estimate obtained using DP is seen to be better than that obtained using local optimization. Due to the large size of the image and the disparities, DP takes a total of 81 seconds for computing the disparity. Using wavelet pyramiding along with narrow-band DP, we are able to reduce this to 43 seconds. The parameters are: $\Omega_p = 5, \Omega_q = 5, B = 60$, and $\lambda = 0.06$.

Part II: Space-variant Filtering

Chapter 8

Fast space-variant Gaussian filtering

Abstract — The efficient realization of linear space-variant (non-convolution) filters is a challenging computational problem in image processing. In this chapter, we demonstrate that it is possible to filter an image with a Gaussian-like elliptic window of varying size, elongation and orientation using a fixed number of computations per pixel. The filtering algorithm is based on a family of compactly supported piecewise-polynomials, and is realized using pre-integration and local finite-differences[1].

8.1 Introduction

THE most commonly used smoothing operator in image processing is the Gaussian filter. As far as isotropic Gaussians are concerned, a fast implementation is achievable simply by decomposing it into two 1-dimensional Gaussian, one along each image axes. The 1-dimensional filters are in turn implemented using efficient recursive algorithms, e.g., the ones proposed by Deriche [78] and Young et al. [79]. We refer the readers to this survey article [80] for an exhaustive account and comparison of such recursive techniques.

A fundamental limitation of isotropic kernels is that it does not take into account the anisotropic nature of image features. This results in the blurring of oriented patterns and textures. The development of fast anisotropic filters, in particular, anisotropic Gaussian filters, have gained momentum over the past decade. Worth

---

[1]This chapter is based on the article [77]: K. N. Chaudhury, A. Muñoz-Barrutia, M. Unser, "Fast space-variant elliptical filtering using box splines," IEEE Transactions on Image Processing, vol. 19, no. 9, pp. 2290-2306, September 2010.

mentioning in this regard is the work of Geusebroek et al. [81], who developed an efficient recursive technique based on the factorization of the anisotropic Gaussian into two 1-dimensional Gaussians, one along the image axes and the other along a generally off-grid direction. A drawback of this technique is that one has to interpolate the image along the off-grid direction to implement the corresponding 1-dimensional filter. To avoid interpolation and, in effect, to improve the spatial homogeneity and the Gaussian structure of the filters in [81], Lam et al. came up with their "triple-axis" filter. Instead of using two directions, they chose to decompose the anisotropic Gaussian into three 1-dimensional Gaussians, operating along one of the four cardinal directions—the horizontal, the vertical, and the two diagonals [82]. The focus of these papers have largely been on space-invariant filtering, where the entire image is convolved with a single anisotropic Gaussian. As against this, a variety of space-variant filtering strategies have also been developed. This included image statistics driven filtering [83], non-linear diffusion filtering [31, 84] and gradient inverse-weighted filtering [85], to name a few.

8.2   Space-variant averaging

In this Chapter, we focus on linear space-variant filtering using Gaussian-like kernels of various shapes and sizes. From a purely discrete perspective, this calls for the design of a family of Gaussian filters $\{g_\lambda[n]\}_\lambda$, so that, given an input image $f[n]$, one gets the filtered samples $\bar{f}[n]$ through the averaging

$$\bar{f}[n] = \sum_{k \in \mathbb{Z}^2} f[k] g_{\lambda(n)}[n - k]. \qquad (8.1)$$

The parameter $\lambda(n)$ specifies the covariance of the filter applied at location n. It allows one to continuously adjust the scale, orientation and elongation of the filter in keeping with the anisotropy of the local image features. There are, however, certain practical problems involved in an efficient realization of (8.1). It is obvious that (8.1) cannot be written as a convolution, and hence cannot be realized using the fast Fourier transform. In fact, the available options are either to (i) compute the filters $g_\lambda[n]$ by sampling the continuous Gaussian on-the-fly, or (ii) discretize λ a priori and to pre-compute the corresponding filters. The problem with the former is that it proves to be extremely slow for wide kernels, while the latter restricts the control on the anisotropy of the filters. By appropriately modifying the algorithm in [78, 79], Tan et al. developed an algorithm for computing (8.1) for the particular case when $g_\lambda[n]$ is isotropic [80].

Spline kernels can also yield efficient algorithms for space-variant filtering, particularly when the space-variance is in terms of the scale (or size) of the kernel. For instance, Heckbert proposed an algorithm for adaptive image blurring using tensor-products of polynomial splines, where the image is filtered with kernels of various scales using repeated integrations and finite-differences [28]. Based on similar principles, namely, the scaling properties of B-splines, Muñoz-Barrutia et

al. have developed an algorithm for rapid computation of the continuous wavelet transform of 1-dimensional signals [86]. Recently, the method was extended to perform space-variant filtering using Gaussian-like functions of arbitrary size [87]. To achieve this, the authors choose to approximate the Gaussian using separable B-splines. We propose to take this approach one step further. In particular, we overcome the limited steerability and ellipticity of the separable B-splines by considering certain quasi-separable analogues of B-splines, the so-called *box splines* [6]. We show that these box splines can also be used to approximate anisotropic Gaussians, and that the associated space-variant filter can be realized using recursive pre-integrations and scale-dependent finite differences.

To date, there have only been few applications of such multivariate splines in image processing and computer graphics. Noteworthy among them are the works of Richter [88] and Asahi et al. [89] on the development of image approximation and reconstruction algorithms, and that of Condat et al. [90] and Entezari et al. [91] on the development of interpolation formulas for hexagonal and BCC lattices.

8.3   The main idea

We propose a fast space-variant filtering algorithm using a family of Gaussian-like box splines whose size, elongation and orientation can be continuously controlled. The attractive feature of our approach is that we use a continuous-discrete formalism which avoids the necessity of sampling a continuously-defined Gaussian on-the-fly, or of storing a discrete set of pre-computed filters. The subsequent developments are centered around the following main ideas.

(1) **The use of quasi-separable box splines**. The construction of bivariate box splines, conceived as the "shadow" of N-dimensional polytopes in 2-dimensions, often turns out to be rather intricate; see [6] for instance. We consider an alternative straightforward recipe for constructing box splines, namely, through repeated convolutions of dilated and rotated box distributions (see Figure 8.3). In particular, we realize the *radially-uniform box spline* $\beta_a^N(x)$ through the convolution of N rotated box distributions, where $a = (a_1, \ldots, a_N)$ is a scale-vector with a_k being the scale of the box distribution along the direction $(k-1)\pi/N$. We give a precise definition in §8.5.

The reason why the radially-uniform box splines are of interest in the current context is twofold. The first of these is that we can make them arbitrarily close to a Gaussian by increasing N. The second reason, which has a more practical significance, is that we can continuously control their size, elongation, and orientation simply by acting on the scales.

(2) **An efficient strategy for space-variant averaging**. To convey this idea, consider the formula

$$\bar{f}[n] = \frac{1}{2W(n)+1} \sum_{k=-W(n)}^{W(n)} f[n-k]. \tag{8.2}$$
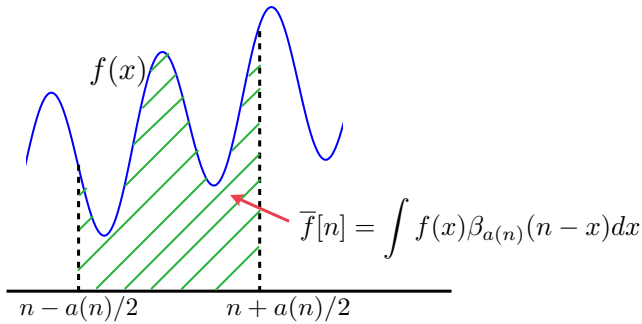
Figure 8.1: Computation of the space-variant average $\bar{f}[n]$. We first
localize the signal $f(x)$ using the shifted box function (hatched zone), and
then we compute the area. The central idea of our algorithm is to determine
this area by taking the finite-difference of the primitive of $f(x)$.

This gives the space-variant averages of a 1-dimensional signal $f[n]$. We interpret the
factor $W(n)$ as the size of the "discrete box filter" applied at location n. It controls
the amount of smoothing. The disadvantage of (8.2) is that the computational cost
scales linearly with $W(n)$, which even gets worse in higher dimensions. This can be
circumvented (with a mild interpolation cost) if we instead consider the formula

$$\bar{f}[n] = \frac{1}{a(n)} \int_{n-a(n)/2}^{n+a(n)/2} f(y)\, dy = \int f(y)\text{Box}_{a(n)}(n-y)\, dy. \qquad (8.3)$$

Here we have replaced the discrete signal $f[n]$ by its interpolation $f(x)$, and the
discrete box filter by the (normalized) box function

$$\text{Box}_a(x) = \begin{cases} 1/a & \text{for } -a/2 < x \le a/2, \\ 0 & \text{otherwise.} \end{cases}$$

The main advantage is that we can realize (8.3) using a fixed number of computations
per position, independent of the size of $a(n)$. To do so, we first compute the
primitive

$$F(x) = \int_{-\infty}^{x} f(y)\, dy$$

and then apply the formula

$$\bar{f}[n] = \frac{1}{a(n)} \Big[F(n+a(n)/2) - F(n-a(n)/2) \Big]. \qquad (8.4)$$

The last step requires one addition and multiplication per position. The idea is
illustrated in Figure 8.1. The other advantage is that, as opposed to the integer-valued

Figure 8.2: Box function rescaling through "addition and subtraction" of the unit box function: The step function is first reproduced from the unit box using the running-sum, and then the appropriate finite-difference is applied to recover the rescaled box function.
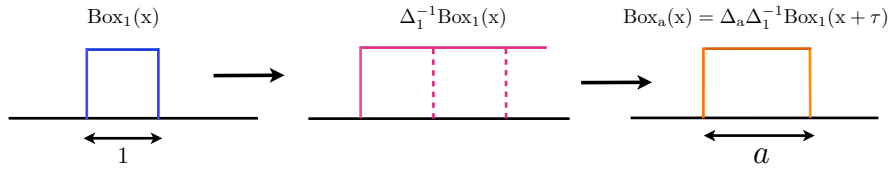
window $W(n)$ in (8.2), this gives access to the real-valued scale $a(n)$ for continuously controlling the smoothing. Indeed, if $f(x)$ is integrable (at least locally), then it can be shown that the use of small scales results in less smoothing. In particular, $\bar{f}[n]$ tends to $f[n]$ as $a(n)$ goes to zero. On the other hand, $\bar{f}[n]$ becomes small as $a(n)$ get large.

Our present contribution is the generalization of (8.3) to the bivariate setting using the radially-uniform box splines. In particular, given a discrete image $f[\boldsymbol{n}]$, we consider the space-variant filtering

$$\bar{f}[\boldsymbol{n}] = \int f(\boldsymbol{y}) \beta^N_{a(\boldsymbol{n})}(\boldsymbol{n} - \boldsymbol{y}) \, d\boldsymbol{y} \tag{8.5}$$

where $f(\boldsymbol{x})$ represents a suitable interpolation of $f[\boldsymbol{n}]$. The significance of the quasi-separable characterization of $\beta^N_a(\boldsymbol{x})$ in terms of the box distributions is that it allows us to relate (8.5) to the 1-dimensional problem in (8.3). Indeed, we demonstrate in §8.5 that (8.5) can be implemented using an appropriate bivariate extension of pre-integrations and finite-differences, together with few evaluations of a fixed piecewise polynomial (the coefficients are pre-computed). Although the derivation of the algorithm is slightly involved, the final solution turns out to be remarkably simple and easy to implement; see §8.6, Algorithm 3.

8.4   Efficient space-variant averaging

We now derive (8.4) using an operator-based formalism. This helps set up the framework required for the subsequent generalization of the idea to higher dimensions and multiple orders.

For positive real numbers a and b, let

$$\Delta_a f(x) = \frac{1}{a}\left[f(x) - f(x-a)\right] \tag{8.6}$$

and

$$\Delta_b^{-1} f(x) = b \sum_{k=0}^{\infty} f(x-bk). \tag{8.7}$$

We call these the *finite-difference* (FD) and the *running-sum* (RS) of $f(x)$, respectively[2]. The numbers a and b are referred to as the *scales* of the operators.

We note that the operators Δ_a and Δ_b^{-1}, which take $f(x)$ into $\Delta_a f(x)$ and $\Delta_b^{-1} f(x)$ respectively, are linear and translation-invariant. Moreover, when b is an integer, Δ_b^{-1} can be applied to a sequence $f[n]$ through the well-defined operation

$$\Delta_b^{-1} f[n] = b \sum_{k=0}^{\infty} f[n-bk].$$

In particular, $g[n] = \Delta_1^{-1} f[n]$ can be efficiently implemented using the recursion $g[n] = g[n-1] + f[n]$, under appropriate boundary conditions [86].

Note that we can relate the variable-size box functions in (8.3) to the unit-width box function using the transformation

$$\mathrm{Box}_a(x) = \Delta_a \Delta_1^{-1} \mathrm{Box}_1(x+\tau) \tag{8.8}$$

where $\tau = (a-1)/2$. This means that box functions of variable widths can be derived from a fixed box function through the successive applications of running-sums and finite-differences; see Figure 8.2. To derive (8.8), we note that

$$\Delta_1^{-1} \mathrm{Box}_1(x) = \sum_{k=0}^{\infty} \mathrm{Box}_1(x-k) = u(x+1/2),$$

where $u(x) = 1$ for $x > 0$, and is zero otherwise. This shows that

$$\Delta_a \Delta_1^{-1} \mathrm{Box}_1(x+\tau) = \frac{1}{a}\left[u(x+a/2) - u(x-a/2)\right] = \mathrm{Box}_a(x).$$

We use (8.8) to derive the algorithm for computing (8.3) as follows. Fix some arbitrary n and the corresponding $a(n)$ in (8.3), and consider the function

$$s(x) = \int f(y)\mathrm{Box}_1(x-y)\,dy.$$

---

[2]The notation Δ_b^{-1} is justified by the fact that $\Delta_a \Delta_b^{-1}$ acts as the identity operator when $a = b$.

By linearity and translation-invariance, and using (8.8), we can write

$$\Delta_{a(n)}\Delta_1^{-1}s(x+\tau) = \int f(y)[\Delta_{a(n)}\Delta_1^{-1}\text{Box}_1(x+\tau-y)]\,dy$$

$$= \int f(y)\text{Box}_{a(n)}(x-y)\,dy.$$

In other words,

$$\bar{f}[n] = \Delta_{a(n)}\Delta_1^{-1}s(n+\tau).$$

Now if the input is discrete, of the form $f(x) = \sum_{n\in\mathbf{Z}} f[n]\delta(x-n)$, then

$$s(x) = \sum f[n]\text{Box}_1(x-n).$$

A simple manipulation then shows that

$$\Delta_1^{-1}s(x) = \sum g[n]\text{Box}_1(x-n)$$

where $g[n]$ is the running-sum of $f[n]$. Therefore, denoting the piecewise-constant interpolation of $g[n]$ by $F(x)$, we obtain

$$\bar{f}[n] = \Delta_{a(n)}F(n+\tau).$$

This leads us to the following two-step algorithm for realizing (8.3).

(1) **Space-invariant step**. Perform the recursion $g[n] = g[n-1] + f[n]$.

(2) **Space-variant step**. Set $\bar{f}[n] = \Delta_{a(n)}F(n+\tau)$, where $\tau = (a(n)-1)/2$.

The steps of the algorithm when the input is an impulse, and when $a(n) = a$ for every n, are shown in Figure 8.2. The second and third plots correspond to steps (1) and (2) of the algorithm.

The remarkable fact about the algorithm is that the space-variant aspect of the transformation $f[n] \mapsto \bar{f}[n]$ is transferred to the scale-dependent operator Δ_a. This is then implemented at a fixed computational cost per pixel—one addition and multiplication.

We note that (8.8) can more generally be written as

$$\text{Box}_a(x) = \Delta_a\Delta_b^{-1}\text{Box}_b(x+\tau) \tag{8.9}$$

where $\tau = (a-b)/2$. The significance is that, if the lattice spacing b is different from unity, one can still realize the running-sum (without interpolation) by replacing the operator Δ_1^{-1} by Δ_b^{-1}. We will use this idea in the sequel.

8.5   Bivariate extension

We now extend the space-variant filtering discussed in the previous section to the bivariate setting. Here, we need to address the additional aspect of directionality.
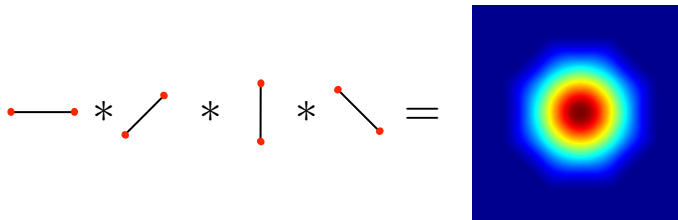
(A)   Four-directional box spline
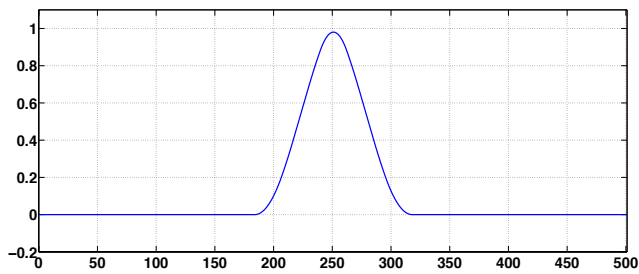
(B)   Scan profile along $\theta = \pi/8$

Figure 8.3: Construction of the radially-uniform box spline through the convolution of four directional box distributions. (**A**) The four box distributions, distributed uniformly over $[0, \pi)$, were assigned equal scales in this example; (**B**) Scan profile along $\theta = \pi/8$.

8.5.1   Radially-uniform box splines

As a first step, we devise an appropriate directional extension of the box function. Corresponding to some real-valued scale a and direction $0 \le \theta < \pi$, we define

$$\varphi_{a,\theta}(\boldsymbol{x}) = \mathrm{Box}_a(\boldsymbol{u}_\theta^T \boldsymbol{x}) \delta(\boldsymbol{u}_{\theta_\perp}^T \boldsymbol{x})$$

where $\boldsymbol{u}_\theta = (\cos\theta, \sin\theta)$ and $\boldsymbol{u}_{\theta_\perp} = (-\sin\theta, \cos\theta)$. We call $\varphi_{a,\theta}(\boldsymbol{x})$ the *directional box distribution*. It is the tensor product of the box function $\mathrm{Box}_a(x)$ and the Dirac distribution $\delta(x)$ operating along orthogonal directions $\boldsymbol{u}_\theta$ and $\boldsymbol{u}_{\theta_\perp}$. The scale a controls the amount of smoothing applied along the direction of the box distribution, whereas, no smoothing is applied along the transverse direction.

The idea then is to convolve an arbitrary number of such directional box distributions (see Figure 8.3). Corresponding to some integer $N \ge 2$, a set of real-valued scales $a_1, \ldots, a_N$, and uniform rotations $\theta_k = (k-1)\pi/N, k = 1, \ldots, N$, we define

$$\beta_a^N(\boldsymbol{x}) = (\varphi_{a_1,\theta_1} * \cdots * \varphi_{a_N,\theta_N})(\boldsymbol{x}). \tag{8.10}$$

We call this the *radially-uniform box spline*. We refer to N as the *directional order* (or simply order), and the tuple $a = (a_1, \ldots, a_N)$ the *scale-vector* of the box spline.

8.5.2   Properties

We note that a more standard (though less intuitive) analytic definition of the radially-uniform box spline is as follows. Let $\mathbf{M}$ be the matrix whose columns are given by $a_k \boldsymbol{u}_{\theta_k}, 1 \le k \le N$, and let $\square = [-1/2, 1/2]^N$ be the centered unit cube (box) in $\mathbf{R}^N$. Consider the distribution

$$\langle B_{\mathbf{M}}, f \rangle = \int_\square f(\mathbf{M}\boldsymbol{x}) d\boldsymbol{x} \tag{8.11}$$

where $f(\boldsymbol{x})$ is some continuous test function. Then $\beta_a^N(\boldsymbol{x})$ is the "density" function associated with $B_{\mathbf{M}}$, that is,

$$\langle B_{\mathbf{M}}, f \rangle = \int f(\boldsymbol{x}) \beta_a^N(\boldsymbol{x}) d\boldsymbol{x}.$$

To see the equivalence between (8.10) and (8.11), we set $f(\boldsymbol{x}) = \exp(-j\boldsymbol{\omega}^T \boldsymbol{x})$ in (8.11), and apply the convolution-multiplication property of the Fourier transform. Moreover, by setting $f(\boldsymbol{x}) = 1$ in (8.11), we see that the total mass of $\beta_a^N(\boldsymbol{x})$ is given by the volume of the unit cube in $\mathbf{R}^N$. This explains why $\beta_a^N(\boldsymbol{x})$ is often described as the "shadow" of a N-dimensional box in 2-dimensions [6].

Following either of the definitions, it can be verified that $\beta_a^N(\boldsymbol{x})$ is a piecewise polynomial of degree $\le N - 2$, where the partitions are specified by lines running

along the directions $\theta_1, \ldots, \theta_N$. It is symmetric with respect to the origin, and is compactly supported on a convex N-sided polygon consisting of the points

$$\left\{ \sum_{k=1}^{N} \alpha_k \, a_k \boldsymbol{u}_{\theta_k} : \; -1/2 \leq \alpha_k \leq 1/2 \right\}.$$

Clearly, the radially-uniform box splines are non-separable for $N > 2$. However, in keeping with the spirit of the underlying tensor construction, the term *quasi-separable* is more appropriate.

The scale-vector plays a vital role in determining the size and shape of the box spline. It is clear that the box spline can be arbitrarily elongated along the principal directions θ_k $(1 \leq k \leq N)$ simply by rescaling the box distribution φ_{a_k,θ_k}, that is, by making a_k large compared to the other scales. In fact, we will demonstrate in the sequel that one can elongate the box spline along *any* arbitrary direction by appropriately acting on the scale-vector. The role of the directional order is more subtle. It determines the degrees of freedom available for controlling the geometry of the box spline and also its smoothness. We will discuss these aspects in detail for the particular four-directional box spline ($N = 4$) in §9.8.

8.5.3   Realization of (8.5)

First, we extend the FD and the RS operator to bivariate functions. Our goal is to derive a relation similar to (8.8) for the radially-uniform box splines. Corresponding to positive real-valued scales a and b, and direction $0 \leq \theta < \pi$, we consider the *directional finite-difference*

$$\Delta_{a,\theta} f(\boldsymbol{x}) = \frac{1}{a} \left[f(\boldsymbol{x}) - f(\boldsymbol{x} - a\boldsymbol{u}_\theta) \right], \tag{8.12}$$

and the *directional running-sum*

$$\Delta_{b,\theta}^{-1} f(\boldsymbol{x}) = b \sum_{k=0}^{\infty} f(\boldsymbol{x} - k b \boldsymbol{u}_\theta). \tag{8.13}$$

The radially-uniform FD and RS operators are then specified by the combined action of (8.12) and (8.13) along the directions $\theta_k = (k-1)\pi/N$. In particular, we set

$$\Delta_{\boldsymbol{a}}^{N} = \Delta_{a_1,\theta_1} \circ \cdots \circ \Delta_{a_N,\theta_N}, \tag{8.14}$$

and

$$\Delta_{\boldsymbol{b}}^{-N} = \Delta_{b_1,\theta_1}^{-1} \circ \cdots \circ \Delta_{b_N,\theta_N}^{-1}, \tag{8.15}$$

where the scale-vectors $\boldsymbol{a} = (a_1, \ldots, a_N)$ and $\boldsymbol{b} = (b_1, \ldots, b_N)$ specify the scale along each direction. The operators $\Delta_{\boldsymbol{a}}^{N}$ and $\Delta_{\boldsymbol{b}}^{-N}$ are closely related to the radially-uniform box splines.

Proposition 27 (The key decomposition). *The box spline $\beta_a^N(x)$ can be expressed as*

$$\beta_a^N(x) = \Delta_a^N \Delta_b^{-N} \beta_b^N(x + \tau).$$ (8.16)

where τ is an appropriate shift vector.

Proof. It is straightforward to be verify that

$$\Delta_a^N \Delta_b^{-N} = \Delta_{a_1,\theta_1} \Delta_{b_1,\theta_1}^{-1} \circ \cdots \circ \Delta_{a_N,\theta_N} \Delta_{b_N,\theta_N}^{-1}.$$ (8.17)

From (8.9), we can write

$$\varphi_{a,\theta}(x) = \Delta_{a,\theta} \Delta_{b,\theta}^{-1} \varphi_{b,\theta}(x + \tau u_\theta).$$ (8.18)

where $\tau = (a - b)/2$.

Let $\tau = \sum \tau_k u_{\theta_k}$, where $\tau_k = (a_k - b_k)/2$. Following definition (8.10), we can write

$$\beta_b^N(x + \tau) = \circledast_{k=1}^N \varphi_{b_k,\theta_k}(x + \tau_k u_{\theta_k}).$$ (8.19)

From (8.17), (8.18), and (8.19), we see that

$$\begin{aligned}
\Delta_a^N \Delta_b^{-N} \beta_b^N(x + \tau) &= \Delta_{a_1,\theta_1} \Delta_{b_1,\theta_1}^{-1} \circ \cdots \circ \Delta_{a_N,\theta_N} \Delta_{b_N,\theta_N}^{-1} \left[\circledast_{k=1}^N \varphi_{b_k,\theta_k}(x + \tau_k u_{\theta_k}) \right] \\
&= \circledast_{k=1}^N \Delta_{a_k,\theta_k} \Delta_{b_k,\theta_k}^{-1} \varphi_{b_k,\theta_k}(x + \tau_k u_{\theta_k}) \\
&= \circledast_{k=1}^N \varphi_{a_k,\theta_k}(x)
\end{aligned}$$

which is simply $\beta_a^N(x)$ by (8.10). $\qquad\square$

Before discussing the filtering algorithm, we briefly elaborate on the implementation of Δ_a^N and Δ_b^{-N}. It can be shown that (8.14) can be written as

$$\Delta_a^N f(x) = \sum_{i=0}^{2^N - 1} w_i f(x - x_i)$$ (8.20)

where

$$w_i = (-1)^{q_1 + \cdots + q_N} (a_1 \cdots a_N)^{-1} \quad \text{and} \quad x_i = \sum_{k=1}^N q_k a_k u_{\theta_k}.$$

The index i is the decimal counterpart of the binary number $(q_N, q_{N-1}, \ldots, q_1)$. It takes values from $0, \cdots, 2^N - 1$ corresponding to the values $(0, \ldots, 0), \cdots, (1, \ldots, 1)$ taken by $(q_N, q_{N-1}, \ldots, q_1)$. We call x_i the vertices of the FD mesh, and w_i the corresponding mesh taps (see Figure 8.5).

As far as the application of Δ_b^{-N} to a discrete sequence $f[n]$ is concerned, the difficulty is that the vectors $b_k u_\theta$ must necessarily lie on the lattice for (8.13) to be

well-defined. Therefore, we cannot associate a digital filter with the RS operators in general. However, the good news is that this can be done when N equals 2 and 4. We will discuss this in detail in §8.6.

The algorithm for realizing (8.5) corresponding to a specified scale-vector map $a(n)$ is based on (8.16). In particular, we assume the input signal to be of the form

$$f(x) = \sum_{n \in \mathbf{Z}^2} f[n] \delta(x - n).$$

The pre-integrated signal is defined to be

$$g_b[n] = \Delta_b^{-N} f[n].$$

By proceeding along lines of the 1-dimensional case, we can write (8.5) as

$$\bar{f}[n] = \sum_{i=0}^{2^N - 1} w_i F(n + \tau - x_i) \tag{8.21}$$

where

$$F(x) = \sum_{n \in \mathbf{Z}^2} g_b[n] \beta_b^N(x - n).$$

and

$$\tau = \frac{1}{2} \Big(\sum (a_k(n) - b_k) \cos \theta_k, \ \sum (a_k(n) - b_k) \sin \theta_k \Big).$$

The pairs (x_i, w_i) are the vertices and taps of the affine FD mesh in (8.20). Note that τ, w_i and x_i are defined pointwise in (8.21); we dropped the index n to simplify the notation. We discuss the implementation aspects of the algorithm, particularly the computation of $g_b[n]$ and its interpolation $F(x)$, for the case $N = 4$ in §8.6.

8.5.4 Asymptotic characterization

The motivation behind introducing the radially-uniform box splines was to develop elliptical Gaussian-like filters, whose shape (size, elongation and orientation) can be continuously controlled. It turns out that the radially-uniform box splines (and their iterated versions) form close approximates of the Gaussian. The following result (see Appendix A for a proof) can be seen as a "radial" version of the central limit theorem[3].

Theorem 28 (Approximation of isotropic Gaussians). *Let $N \geq 2$, and set*

$$a_k(N) = \sigma \sqrt{\frac{24}{N}} \qquad (1 \leq k \leq N).$$

---

[3]An interpretation of this principle in the probabilistic setting is discussed in Appendix G.

Then

$$\lim_{N \longrightarrow \infty} \beta^N_{a(N)}(x) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\|x\|^2}{2\sigma^2}\right). \tag{8.22}$$

The radially-uniform box splines constructed using uniform scale-vectors are supported on a N-sided uniform polygon, and it can be shown that they have continuous derivatives of order $(N-3)$. The above result is then consistent with the fact that the isotropy and smoothness of such box splines progressively improves with the increase in the directional order. It is also possible to mimic certain anisotropic Gaussians by using a sequence of non-uniform scale-vectors. Indeed, as a direct extension of Theorem 28, one can construct sequences of box splines which converge to anisotropic Gaussians as N increases.

Another useful form of anisotropic convergence is achievable based on the serial convolutions of a radially-uniform box spline (fixed directional order) with itself. In particular, corresponding to fixed integers N and m ($m \geq 1$), and a scale-vector $a = (a_1, \ldots, a_N)$, consider the *iterated radially-uniform box spline*

$$\beta^{N,m}_a(x) = (\beta^N_a * \cdots * \beta^N_a)(x) \tag{8.23}$$

obtained through the $(m-1)$-fold convolution of $\beta^N_a(x)$ with itself. Let

$$a(m) = \frac{1}{\sqrt{m}}(a_1, \ldots, a_N).$$

Then

$$\lim_{m \longrightarrow \infty} \beta^{N,m}_{a(m)}(x) = \frac{1}{2\pi\,|\det(\mathbf{C})|^{1/2}} \exp\left(-\frac{1}{2}x^T\mathbf{C}^{-1}x\right) \tag{8.24}$$

where

$$\mathbf{C} = \frac{1}{12}\begin{pmatrix} \sum a_k^2 \cos^2\theta_k & \frac{1}{2}\sum a_k^2 \sin 2\theta_k \\ \frac{1}{2}\sum a_k^2 \sin 2\theta_k & \sum a_k^2 \sin^2\theta_k \end{pmatrix}. \tag{8.25}$$

To see this, note that the components of the box spline are given by

$$\circledast^m_{j=1} \varphi_{a_k/\sqrt{m},\theta_k}(x) \qquad (1 \leq k \leq N).$$

It follows from a certain from of the central limit theorem that each of these converge to a "directional" Gaussian distribution as m gets large. The covariance in (8.25) is the limiting sum of the covariances of the constituent box distributions. The utility of these iterated box splines will be discussed in §8.7.

8.5.5 Approximation of Gaussian

Having characterized the asymptotic behavior of the box splines, we now focus on the problem of approximating a given anisotropic Gaussian using box splines
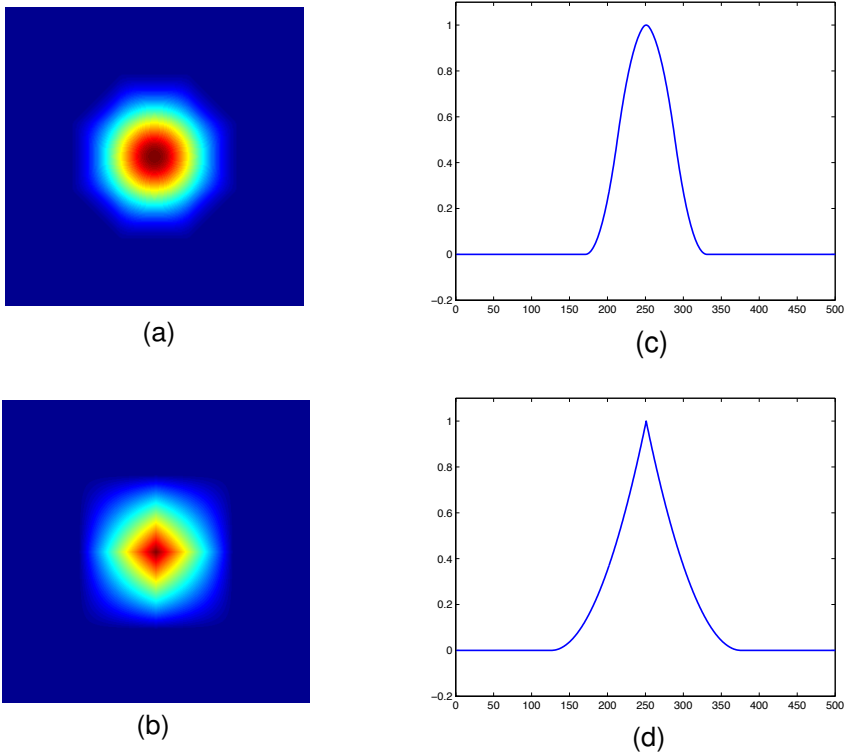
Figure 8.4: Intensity distribution of (a) the radially-uniform box spline, and (b) the separable B-spline, of order four. The respective scan profiles along $\pi/8$ are shown in (c) and (d).

of fixed directional order. Since a centered Gaussian is uniquely specified by its covariance, we propose to approximate the Gaussian based on its covariance. Since the level-sets of Gaussians are ellipses, this is also equivalent to constructing elliptical filters of different size, elongation, and orientation.

The covariance of the radially-uniform box spline is given by

$$\mathbf{C}_a^N = \int x x^T \beta_a^N(x)\, dx.$$

As shown in Appendix B , we can express this as the sum of the covariances of the box distributions,

$$\mathbf{C}_a^N = \frac{1}{12} \begin{pmatrix} \sum a_k^2 \cos^2\theta_k & \frac{1}{2}\sum a_k^2 \sin 2\theta_k \\ \frac{1}{2}\sum a_k^2 \sin 2\theta_k & \sum a_k^2 \sin^2\theta_k \end{pmatrix}. \tag{8.26}$$

This gives the explicit dependence of $\mathbf{C}_a^N$ on the scale-vector. We propose the following characterization of the elliptical parameters of the box spline: Let $\lambda_{\max}$ and $\lambda_{\min}$ denote the largest and smallest eigenvalues of $\mathbf{C}_a^N$, and (v_1, v_2) the eigenvector corresponding to the eigenvalue $\lambda_{\max}$. The size, elongation, and orientation of the radially-uniform box spline are then defined by

$$s_a^N = \lambda_{\max} + \lambda_{\min} = \frac{1}{12}\sum a_k^2,$$

$$\varrho_a^N = \frac{\lambda_{\max}}{\lambda_{\min}} = \frac{\sum a_k^2 + \sqrt{D}}{\sum a_k^2 - \sqrt{D}}, \text{ and}$$

$$\theta_a^N = \tan^{-1}\left(\frac{v_2}{v_1}\right) = \tan^{-1}\left[\frac{-\sum a_k^2 \cos(2\theta_k) + \sqrt{D}}{\sum a_k^2 \sin(2\theta_k)}\right], \tag{8.27}$$

where

$$D = \left(\sum a_k^2 \cos 2\theta_k\right)^2 + \left(\sum a_k^2 \sin 2\theta_k\right)^2.$$

Since $\mathbf{C}_a^N$ is strictly positive (again see Appendix B), all the above parameters are indeed well-defined. Note that the covariance in (8.26) and the triple in (8.27) provide equivalent descriptions of the box spline geometry. The motivation behind introducing the latter is its convenient rotation-invariant nature: while $\mathbf{C}_a^N$ changes with the rotations of a given box spline, s_a^N and ϱ_a^N remains fixed.

8.6 Four-directional box splines

We now focus on the *four-directional* box spline defined by

$$\beta_a^4(x) = (\varphi_{a_1,0} * \varphi_{a_2,\pi/4} * \varphi_{a_3,\pi/2} * \varphi_{a_4,3\pi/4})(x).$$
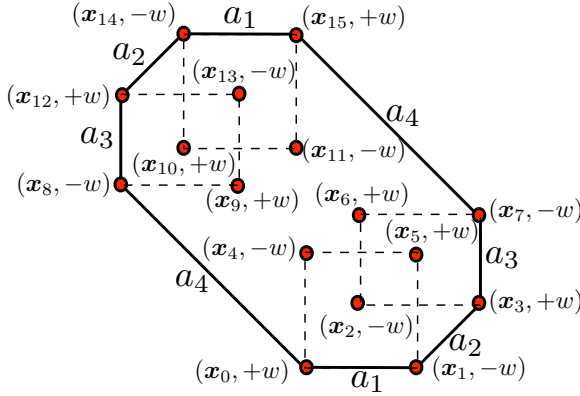
Figure 8.5: The distribution of the taps of the FD mesh. The pairs (u, v) denote the positions (u) and the corresponding weights (v) of the taps of the FD mesh.

This particular box spline is composed of patches of quadratic polynomials (degree ≤ 2), and is continuously differentiable. It is supported on a convex octagon (see Figure 8.3).

We note that in [87] the authors have used separable B-splines to approximate the Gaussian. Although these functions are built from the same constituent box distributions, the advantage of the four-directional box splines over the separable ones is that they are more isotropic. As seen in Figure 8.4, they have a better Gaussian profile than their separable counterparts having identical order. In addition, the anisotropic four-directional box spline can be rotated to arbitrary orientations, while the separable ones are constrained to the image axes.

8.6.1   Fast space-variant elliptical filtering

Note that, for $b = (1, \sqrt{2}, 1, \sqrt{2})$, the running-sum $\Delta_b^{-4} f[n]$ can be computed without interpolation . The corresponding interpolating function, $\beta_b^4(x)$, turns out to be well-known in the box spline community, and is popularly referred to as the Zwart-Powell (ZP) element [6, 92]. We shall simply denote it by $\beta_{\mathrm{ZP}}(x)$.

We realize (8.5) in two-steps as follows.

(1) **Pre-integration**. We fix $b = (1, \sqrt{2}, 1, \sqrt{2})$. Then

$$\Delta_b^{-4} = \Delta_{1,0}^{-1} \circ \Delta_{\sqrt{2}, \pi/4}^{-1} \circ \Delta_{1, \pi/2}^{-1} \circ \Delta_{\sqrt{2}, 3\pi/4}^{-1}.$$

Then $g_b[\boldsymbol{n}] = \Delta_b^{-4} f[\boldsymbol{n}]$ can be computed as follows.

(RS1) Horizontal running-sum:

$$g_0[n_1, n_2] = \Delta_{1,0}^{-1} f[n_1, n_2] = \sum_{k=0}^{\infty} f[n_1 - k, n_2].$$

(RS2) First-diagonal running-sum:

$$g_{\pi/4}[n_1, n_2] = \Delta_{\sqrt{2}, \pi/4}^{-1} g_0[n_1, n_2] = \sqrt{2} \sum_{k=0}^{\infty} g_0[n_1 - k, n_2 - k].$$

(RS3) Vertical running-sum:

$$g_{\pi/2}[n_1, n_2] = \Delta_{1, \pi/2}^{-1} g_{\pi/4}[n_1, n_2] = \sum_{k=0}^{\infty} g_{\pi/4}[n_1, n_2 - k].$$

(RS4) Second-diagonal running-sum:

$$g_b[n_1, n_2] = \Delta_{\sqrt{2}, 3\pi/4}^{-1} g_{\pi/2}[n_1, n_2] = \sqrt{2} \sum_{k=0}^{\infty} g_{\pi/2}[n_1 + k, n_2 - k].$$

(2) **Finite-differences.** At each position $\boldsymbol{n}$, the FD mesh is computed using the scale-vector $\boldsymbol{a}(\boldsymbol{n})$. The weights w_i and the vertices $\boldsymbol{x}_i$ are listed in Table 8.1, where we have set $a_k' = a_k / \sqrt{2}$ for $k = 2$ and 4. The mesh has a total of $4 \times 4 = 16$ vertices (see Figure 8.5). The shift $\tau = (\tau_1, \tau_2)$ is given by

$$\tau_1 = \frac{1}{2\sqrt{2}} (\sqrt{2} a_1 + a_2 - a_4 - \sqrt{2}) \quad \text{and} \quad \tau_2 = \frac{1}{2\sqrt{2}} (a_2 + \sqrt{2} a_3 + a_4 - 3\sqrt{2}).$$

The filtered sample is given by

$$\bar{f}[\boldsymbol{n}] = \sum_{i=0}^{15} w_i F(\boldsymbol{n} + \tau - \boldsymbol{x}_i) \tag{8.28}$$

where

$$F(\boldsymbol{x}) = \sum_{\boldsymbol{n} \in \mathbf{Z}^2} g_b[\boldsymbol{n}] \beta_{\mathrm{ZP}}(\boldsymbol{x} - \boldsymbol{n}).$$

The interpolated samples in (8.28) are computed efficiently by taking advantage of the piecewise-polynomial structure of the compactly supported ZP element (see Appendix F).

As in the 1-dimensional setting, the running-sums are efficiently computed using recursions; see Algorithm 3. The computational advantage, especially for wider kernels, is derived from the fact that the number of vertices of the FD mesh does not depend on the scale-vector. As a result, the algorithm has a fixed computational cost per pixel, modulo the cost of the running-sum and the interpolations (see computation times in Table 8.2).

Table 8.1: Specification of the taps of the FD mesh associated with the operator Δ_a^4. The weight w is given by $(a_1 a_2 a_3 a_4)^{-1}$, where $\boldsymbol{a} = (a_1, a_2, a_3, a_4)$ is the corresponding scale-vector.

i	$\boldsymbol{x}_i$	w_i	i	$\boldsymbol{x}_i$	w_i
0	$(0,0)$	$+w$	8	$(-a_4', a_4')$	$-w$
1	$(a_1, 0)$	$-w$	9	$(a_1 - a_4', a_4')$	$+w$
2	(a_2', a_2')	$-w$	10	$(a_2' - a_4', a_2' + a_4')$	$+w$
3	$(a_1 + a_2', a_2')$	$+w$	11	$(a_1 + a_2' - a_4', a_2' + a_4')$	$-w$
4	$(0, a_3)$	$-w$	12	$(-a_4', a_3 + a_4')$	$+w$
5	(a_1, a_3)	$+w$	13	$(a_1 - a_4', a_3 + a_4')$	$-w$
6	$(a_2', a_3 + a_2')$	$+w$	14	$(a_2' - a_4', a_3 + a_2' + a_4')$	$-w$
7	$(a_1 + a_2', a_3 + a_2')$	$-w$	15	$(a_1 + a_2' - a_4', a_3 + a_2' + a_4')$	$+w$

8.6.2 Size, elongation and orientation of the box splines

The size and shape of the radially-uniform box spline can be controlled by adjusting the scales of the constituent box distributions. In this regard, we now discuss the following: (i) the forward problem of controlling the anisotropy of the four directional box spline by acting on the scale-vector, and (ii) the inverse problem of uniquely specifying the scale-vector of the box spline corresponding to a given covariance (geometry). For notational ease, we henceforth drop the superscript $N = 4$ when referring to the four-directional box spline and its related parameters.

Control on the anisotropy

From (8.27), the size, orientation and elongation this box spline are given by

$$s_a = \frac{1}{12} \sum a_k^2, \quad \theta_a = \tan^{-1}\left[\frac{a_3^2 - a_1^2 + \sqrt{D}}{a_2^2 - a_4^2}\right], \quad \text{and} \quad \varrho_a = \frac{\sum a_k^2 + \sqrt{D}}{\sum a_k^2 - \sqrt{D}},$$

where $D = (a_3^2 - a_1^2)^2 + (a_2^2 - a_4^2)^2$. It turns out that the size and orientation can be arbitrarily controlled by adjusting the scale-vector. Indeed, the size can be easily manipulated by multiplying the scale-vector by a constant factor, since this leaves both the orientation and elongation unchanged. On the other hand, the elongation can be arbitrarily controlled in the neighborhood of the four principal directions. However, there exists a finite upper bound on the elongation along other directions.

Proposition 29 (Feasibilty). *For every ϕ in $[0, \pi)$, there exists a scale-vector a such that $\theta_a = \phi$. There is however a finite bound on the elongation given by*

$$\sup \varrho_a < U(\phi) = \frac{1 + |v_\phi| + \sqrt{1 + v_\phi^2}}{1 + |v_\phi| - \sqrt{1 + v_\phi^2}} \tag{8.29}$$

Algorithm 3 Space-variant box spline filtering

1. Input: $f[\boldsymbol{n}]$ and $\boldsymbol{a}(\boldsymbol{n})$
2. Perform recursions:

$$g_0[n_1, n_2] \leftarrow f[n_1, n_2] + g_0[n_1 - 1, n_2]$$

$$g_{\pi/4}[n_1, n_2] \leftarrow \sqrt{2}g_0[n_1, n_2] + g_{\pi/4}[n_1 - 1, n_2 - 1]$$

$$g_{\pi/2}[n_1, n_2] \leftarrow g_{\pi/4}[n_1, n_2] + g_{\pi/2}[n_1, n_2 - 1]$$

$$g_b[n_1, n_2] \leftarrow \sqrt{2}g_{\pi/2}[n_1, n_2] + g_b[n_1 + 1, n_2 - 1]$$

3. Local FD operation:
for each position $\boldsymbol{n}$ **do**
    Compute w_i, x_i and τ using $\boldsymbol{a}(\boldsymbol{n})$
    Evaluate the samples $F(\boldsymbol{n} + \tau - x_i)$ using ZP interpolation
    $\bar{f}[\boldsymbol{n}] \leftarrow \sum_i w_i F(\boldsymbol{n} + \tau - x_i)$
end for
4. Return $\bar{f}[\boldsymbol{n}]$

---

where $v_\phi = \frac{1}{2}(\tan\phi - \cot\phi)\mathrm{sign}(\frac{\pi}{2} - \phi)$. The supremum is over the set of $\boldsymbol{a}$ for which $\theta_a = \phi$.

This is explained in detail in Appendix C. Figure 8.6 gives the variation of $1/U(\phi)$ as a function of the orientation. In particular, a bound of $3 + 2\sqrt{2} \approx 5.8$ is attained along the orientations $\phi = (2k - 1)\pi/8, 1 \le k \le 4$, exactly mid-way between two adjacent primal directions. This is perfectly reasonable since the control on the geometry of the box spline is minimal along these directions.

In order to specify the elliptical geometry of the box spline, we will use either of the following equivalent descriptors as per convenience:

(D1) Size, elongation and orientation (s, ϱ, θ).

(D2) Length of the major and minor axes, and the orientation $(\sqrt{\lambda_{\max}}, \sqrt{\lambda_{\min}}, \theta)$.

(D3) Covariance matrix **C**.

Descriptor (D1) gives the lengths of the major and minor axes as $\lambda_{\max} = s\varrho/(1 + \varrho)$ and $\lambda_{\min} = s/(1 + \varrho)$, respectivelyyy. On the other hand, (D2) gives the corresponding covariance as

$$\mathbf{C} = \begin{pmatrix} \lambda_{\max}\cos^2\theta + \lambda_{\min}\sin^2\theta & \frac{1}{2}(\lambda_{\max} - \lambda_{\min})\sin 2\theta \\ \frac{1}{2}(\lambda_{\max} - \lambda_{\min})\sin 2\theta & \lambda_{\min}\cos^2\theta + \lambda_{\max}\sin^2\theta \end{pmatrix}.$$
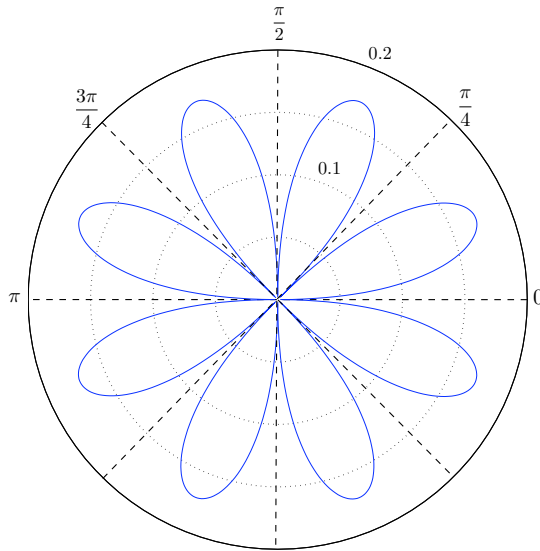
Figure 8.6: Polar plot of the symmetric variation of $1/U(\phi)$ as a function of the filter orientation ϕ, where $U(\phi)$ is the bound on the elongation. We plot the inverse $1/U(\phi)$ to avoid the blowups $U(\phi) \longrightarrow +\infty$ as $\phi \longrightarrow \theta_k$. The bound reaches its minimum when the orientation of the filter is exactly midway between two principal axes, whereas arbitrary elongation is achievable in the neighborhood of the four principal directions $\phi = 0, \pi/4, \pi/2$ and $3\pi/4$.

Optimal scale-vector for a given anisotropy

By (8.26), the covariance matrix of the four-directional box spline is given by

$$\mathbf{C}_a = \frac{1}{24} \begin{pmatrix} 2a_1^2 + a_2^2 + a_4^2 & a_2^2 - a_4^2 \\ a_2^2 - a_4^2 & 2a_3^2 + a_2^2 + a_4^2 \end{pmatrix}. \tag{8.30}$$

The problem is one of specifying a scale-vector a such that $\mathbf{C}_a = \mathbf{C}$. By introducing the positive vector $p = (a_1^2, a_2^2, a_3^2, a_4^2)$, we can reformulate the problem as one of finding $p > 0$ such that $\mathbf{M}p = c$, where

$$\mathbf{M} = \begin{pmatrix} 2 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 1 & 2 & 1 \end{pmatrix} \quad \text{and} \quad c = 24\big(\mathbf{C}(1,1), \mathbf{C}(1,2), \mathbf{C}(2,2)\big).$$

The scale-vector is then given by $a = \sqrt{p}$.

As far as existence of solutions is concerned, Proposition 29 ensures that the linear system $\mathbf{M}p = c$, $p > 0$, corresponding to a given geometry $(\lambda_{\min}, \lambda_{\max}, \theta)$, is always solvable provided that $\varrho < U(\theta)$. As it turns out, the linear system is underdetermined and has infinitely many solutions. The idea then is to use a scale-vector that is "optimal" in some sense.

For reasons that will soon be obvious, we modify the positivity constraint as $p \geq \varepsilon \mathbf{1}$, where ε is some arbitrarily small positive number. Note that $\mathbf{M}$ is of full-rank, and hence the null-space is of dimension $4 - 3 = 1$. This means that the solutions of $\mathbf{M}p = c$ lie on the affine subspace $\{\bar{p} + t\mathbf{e} : t \in \mathbf{R}\}$, where $\bar{p}$ is a particular solution of $\mathbf{M}p = c$, and $\mathbf{e}$ is in the null-space. We fix $\mathbf{e} = (1, -1, 1, -1)$. On the other hand, we can easily compute $\bar{p}$ by pivoting one of its components and solving for the remaining three—since $\mathbf{M}$ is of full-rank, the reduced system is always solvable.

One can easily verify that for $\bar{p} + t\mathbf{e} = (\bar{p}_1 + t, \bar{p}_2 - t, \bar{p}_3 + t, \bar{p}_4 - t) \geq \varepsilon \mathbf{1}$ it is both necessary and sufficient that t lies in the closed interval $[t_\ell, t_r]$, where $t_\ell = \max(-\bar{p}_1 + \varepsilon, -\bar{p}_3 + \varepsilon)$ and $t_r = \min(\bar{p}_2 - \varepsilon, \bar{p}_4 - \varepsilon)$. We use the available degree of freedom to select a solution that maximizes a certain measure of Gaussianity.

A classical measure of the Gaussianity of a 1-dimensional function is its kurtosis.. For a centered function, this is defined by $\varkappa = \mu_4 - 3\mu_2^2$, where μ_4 and μ_2 are the fourth-order and second-order moments. The absolute value of $\varkappa$ provides a measure of Gaussianity. For a true Gaussian, $\varkappa = 0$. In general, the smaller is the absolute value, the more Gaussian-like is the function.

For a bivariate function $f(x)$, we use the following matrix-valued extension

$$\mathbf{K} = \mathbf{L} - \text{tr}(\mathbf{C})\mathbf{C} - 2\mathbf{C}^2 \tag{8.31}$$

where

$$\mathbf{C} = \int (xx^T) f(x) \, dx \quad \text{and} \quad \mathbf{L} = \int (xx^T)^2 f(x) \, dx.$$
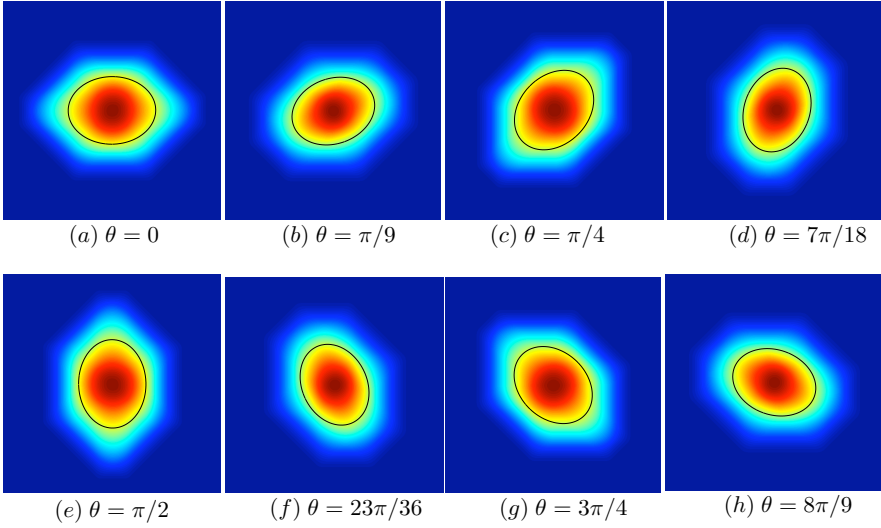
Figure 8.7: Intensity distributions of the four-directional box splines of identical size ($s = 1$) and elongation ($\varrho = 2.5$), but with different orientations. The ellipse in each figure represents a level-set of the Gaussian having the same covariance as the corresponding box spline.

These are the second-order and fourth-order moment matrices of $f(x)$. When $d = 1$, this reduces to $\varkappa = \mu_4 - 3\mu_2^2$. Moreover,

(1) If $f(x)$ is a multivariate Gaussian, then $\mathbf{K} = \mathbf{0}$ (cf. [93] for a proof).

(2) The Frobenius norm of $\mathbf{K}$,

$$\|\mathbf{K}\| = \left(\sum_{i,j} |\mathbf{K}(i,j)|^2 \right)^{1/2},$$

is rotation-invariant in that the kurtosis matrices of the rotations of $f(x)$ have the same Frobenius norm (proof in Appendix D).

Following the above arguments, we propose to solve the optimization problem

$$p_0 = \operatorname{argmin}_p \|\mathbf{K}_p\|^2, \qquad \mathbf{M}p = c, \ p \geq \varepsilon \mathbf{1}. \tag{8.32}$$

This gives the optimal scale-vector $a_0 = \sqrt{p_0}$ corresponding to the most Gaussian-like box spline. The rotation-invariance property ensures that the box splines of identical size and elongation, but of different orientations, are as homogenous as possible.

The norm of the kurtosis matrix of $\beta_a(x)$ is calculated to be

$$\|\mathbf{K}_p\|^2 = \sum_{k=1}^{4} p_k^4 + (p_1^2 + p_3^2)(p_2^2 + p_4^2).$$

See Appendix E for details. Substituting $p_k = \bar{p}_k + te_k$ into this expression, we get the quartic polynomial

$$\zeta(t) = \sum_{k=1}^{4} (\bar{p}_k + e_k t)^4 + \left[(\bar{p}_1 + t)^2 + (\bar{p}_3 + t)^2\right]\left[(\bar{p}_2 - t)^2 + (\bar{p}_4 - t)^2\right].$$

Together with the parameterization $p = \bar{p} + t\mathbf{e}$, this simplifies the problem to one of finding

$$t_0 = \operatorname{argmin}_t \zeta(t), \quad t \in [t_\ell, t_r]. \tag{8.33}$$

The optimal solution is then given by $a_0 = \sqrt{\bar{p} + t_0\mathbf{e}}$.

Note that the minimum of $\zeta(t)$ is attained either at one of the interior points (t_ℓ, t_r) where $\zeta'(t) = 0$, or at one of the boundary points. This gives us a simple algorithm for designing optimized Gaussian-like box splines of a specified covariance. This is shown in Algorithm 4.

The main step is clearly that of root-finding. In particular, the coefficients of the cubic equation $\zeta_1 t^3 + \zeta_2 t^2 + \zeta_3 t + \zeta_4 = 0$ are given by

$\zeta_1 = 32,$
$\zeta_2 = 24(\bar{p}_1 - \bar{p}_2 + \bar{p}_3 - \bar{p}_4),$
$\zeta_3 = 16 \sum \bar{p}_k^2 - 8(\bar{p}_1 + \bar{p}_3)(\bar{p}_2 + \bar{p}_4),$
$\zeta_4 = 4(\bar{p}_1^3 - \bar{p}_2^3 + \bar{p}_3^3 - \bar{p}_4^3) + 2(\bar{p}_1 + \bar{p}_3)(\bar{p}_2^2 + \bar{p}_4^2) - 2(\bar{p}_2 + \bar{p}_4)(\bar{p}_1^2 + \bar{p}_3^2).$

The box splines obtained using the above optimization for various orientations are shown in Figure 8.7. The quality of the Gaussian approximation under different practical settings of the orientation and the elongation is quantified in Figure 8.8.

Algorithm 4 Specification of box spline of given covariance

---

    1. **input**: Covariance triple c.
    2. Set $p_4 = 1$, compute $\bar{p}$ by solving $\mathbf{M}\bar{p} = c$.
    3. Compute t_ℓ and t_r, and the coefficients of $\zeta'(t)$, from $\bar{p}$.
    4. Find the real roots R of $\zeta'(t) = 0$ over the open interval (t_ℓ, t_r).
    5. Set $a_0 = (\bar{p} + t_0\mathbf{e})^{1/2}$, where $\zeta(t_0) = \min_t \zeta(t)$ over the points $R \cup \{t_\ell, t_r\}$.
    6. **return**: The scale vector a_0.

Finally note that the correspondences $(1, \varrho, \theta) \leftrightarrow (a_1, a_2, a_3, a_4)$ for $0 < \theta < \pi$ and $1 \leq \varrho < U(\theta)$ can be pre-computed and stored in a look-up table (LUT). For a given ϱ, the set of correspondences $(1, \varrho, \theta) \leftrightarrow (a_1, a_2, a_3, a_4)$ have an inherent four-fold
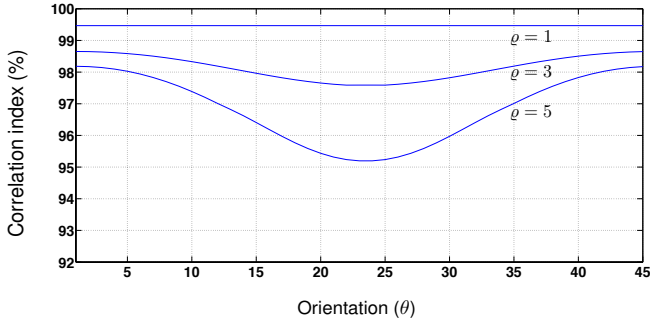
Figure 8.8: Normalized correlation between the optimal four-directional box spline and the target Gaussian at different elongations and orientations. For a fixed elongation, the correlation is minimum at the critical orientation $\theta = 22.5°$, and improves symmetrically as θ approaches the principal orientations (cf. Figure 8.6).

symmetry in θ owing to the presence of the four principal directions. Hence, it suffices to store the scale-vector correspondences for $0 < \theta < \pi/4$. This reduces the size of the LUT by a factor of four. For any arbitrary size $s > 1$, orientation $0 < \theta < \pi$, and elongation $1 \leq \varrho < U(\theta)$, the corresponding scale-vector is given by the following operations.

(1) Rotate the box spline by setting

$$\phi = \begin{cases} \theta & \text{for} \quad 0 < \theta < \pi/4 \\ \theta - \pi/4 & \text{for} \quad \pi/4 < \theta < \pi/2 \\ \theta - \pi/2 & \text{for} \quad \pi/2 < \theta < 3\pi/4 \\ \theta - 3\pi/4 & \text{for} \quad 3\pi/4 < \theta < \pi. \end{cases}$$

(2) Determine (a_1, a_2, a_3, a_4) corresponding to $(1, \varrho, \phi)$ using the LUT. The desired scale-vector is then given by a permutation and rescaling,

$$(a_1, a_2, a_3, a_4) \mapsto \begin{cases} \sqrt{s}(a_1, a_2, a_3, a_4) & \text{for} \quad 0 < \theta < \pi/4 \\ \sqrt{s}(a_2, a_3, a_4, a_1) & \text{for} \quad \pi/4 < \theta < \pi/2 \\ \sqrt{s}(a_3, a_4, a_1, a_2) & \text{for} \quad \pi/2 < \theta < 3\pi/4 \\ \sqrt{s}(a_4, a_1, a_2, a_3) & \text{for} \quad 3\pi/4 < \theta < \pi. \end{cases}$$

8.6.3   Computation time

The space-variant filtering using the four-directional box spline was implemented in Java on a 2.66 GHz Intel system. The typical execution times required for

convolving a 512×512 image with kernels of various sizes are shown in Table 8.2. It is clear that the run time is independent of the size of the kernel.

Table 8.2: Average computation time for box splines of different sizes.

Size (s)	1	2	4	8	16
Time (milliseconds)	101	100	103	101	100

8.7 Higher-order box splines

As suggested by the convergence result (8.22), the Gaussian approximation of the four-directional box splines can be improved by using more directions. Implementing the corresponding space-variant filtering using the algorithm in §8.5.3 however turns out to be challenging and not very practical—the principal axes of these box splines are generally along off-grid directions, and one needs to interpolate the image for implementing the associated running-sums.

The iterated four-directional box splines $\beta_a^{4,m}(x)$ introduced in §9.8 provide a practical alternative. These box splines rapidly converge to a Gaussian with the increase in m. Also, note that the four-directional box spline and its iterates have identical covariances. This means that the algorithm in §8.6.2 can be used for optimizing the iterated box splines as well. The first two iterates of the four-directional box spline along with the target Gaussian are shown in Figure 8.9. It is seen that $\beta_a^{4,2}(x)$ resembles the target Gaussian very closely. In fact, the minimum correlation coefficient rises from 95% to 99% for $m = 2$ (cf. Figure 8.8). In practice, we can thus implement a higher-order filter simply by iterating the algorithm in §8.6.1. We must, of course, set the scale-vector to $a/\sqrt{m}$, where m is the number of iterations.

Appendix A

We first establish that the Fourier sequence $\widehat{\beta}_{a(2)}^2(\omega), \widehat{\beta}_{a(3)}^3(\omega), \dots$ convergences pointwise to a Gaussian:

$$\lim_{N \longrightarrow \infty} \widehat{\beta}_{a(N)}^N(\omega) = \exp\left(-\frac{\sigma^2}{2}\|\omega\|^2\right). \tag{8.34}$$

We then show that the above convergence is also in the $\mathbf{L}^2(\mathbf{R}^2)$ norm. This will establish the theorem, since it is well-known that the Fourier transform of a Gaussian is a Gaussian, and that $f_n \longrightarrow g$ in $\mathbf{L}^2$ if $\hat{f}_n \longrightarrow \hat{g}$ in $\mathbf{L}^2$.

To derive (8.34), we note that

$$\widehat{\varphi}_{a,\theta}(\omega) = \widehat{\beta}_a(u_\theta^T \omega) = \text{sinc}\left(au_\theta^T \omega/2\right)$$
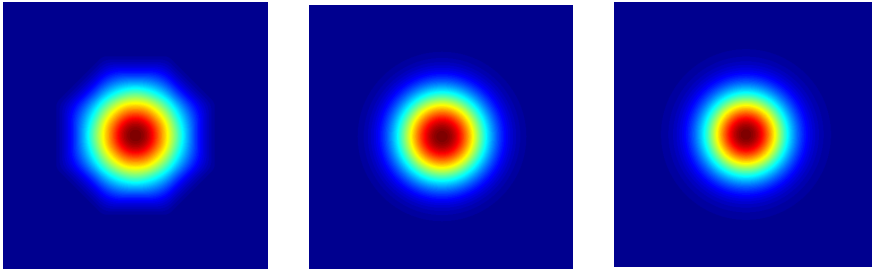
(a) Isotropic forms ($s = 1, \varrho = 1$)
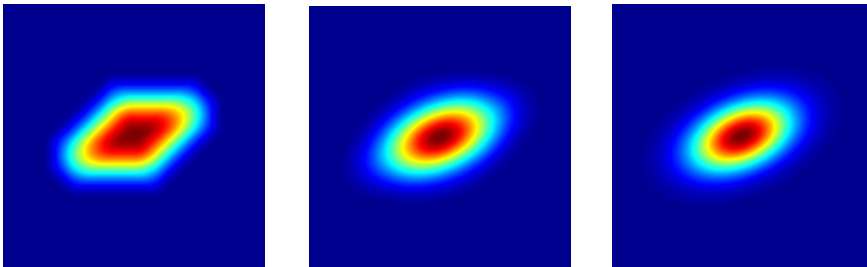
(b) Anisotropic forms ($s = 1, \varrho = 3, \theta = \pi/8$)

Figure 8.9: Higher-order box splines through iterative convolutions. **Left**: The reference four-directional box spline; **Center**: Iterated box spline obtained by convolving the (rescaled) four-directional box spline with itself; **Right**: Target Gaussian having identical covariance.

Then, the convolution-multiplication rule gives

$$\widehat{\beta}_{a(N)}^{N}(\boldsymbol{\omega}) = \prod_{k=1}^{N} \widehat{\varphi}_{a_k(N),\theta_k}(\boldsymbol{\omega}) = \prod_{k=1}^{N} \operatorname{sinc}\left(\frac{a_k(N)}{2}\boldsymbol{u}_{\theta_k}^{T}\boldsymbol{\omega}\right). \tag{8.35}$$

Now, it can be shown that

$$\operatorname{sinc}(x) = 1 - x^2/6 + O(x^4).$$

Substituting $a_k(N) = \sigma\sqrt{24/N}$ into (8.35), and using the above estimate along with the inequality $|\boldsymbol{u}_{\theta_k}^{T}\boldsymbol{\omega}| \leq \|\boldsymbol{\omega}\|$, we have

$$\widehat{\beta}_{a(N)}^{N}(\boldsymbol{\omega}) = \prod_{k=1}^{N}\left[1 - \frac{\sigma^2}{N}(\boldsymbol{u}_{\theta_k}^{T}\boldsymbol{\omega})^2 + O\left(N^{-2}\|\boldsymbol{\omega}\|^4\right)\right]. \tag{8.36}$$

We next develop the quadratic term in (9.6),

$$(\boldsymbol{u}_{\theta_k}^{T}\boldsymbol{\omega})^2 = (\cos\theta_k\omega_1 + \sin\theta_k\omega_2)^2,$$

and then the product. After some calculation, it is found that, for some absolute constant c,

$$\widehat{\beta}_{a(N)}^{N}(\boldsymbol{\omega}) = \prod_{k=1}^{N}\left[1 - \frac{\sigma^2}{2N}\|\boldsymbol{\omega}\|^2 + \frac{\sigma^2}{2N}(\omega_1^2 - \omega_2^2)\cos 2\theta_k + \frac{\sigma^2}{N}\omega_1\omega_2\sin 2\theta_k + O\left(N^{-2}\|\boldsymbol{\omega}\|^4\right)\right]$$

$$= \left(1 - \frac{\sigma^2}{2N}\|\boldsymbol{\omega}\|^2\right)^{N} + O\left(N^{-1}\|\boldsymbol{\omega}\|^4\right) \qquad (\|\boldsymbol{\omega}\| < cN). \tag{8.37}$$

In the above calculation, we have made decisive use of the fact that rotations $\theta_k = (k-1)\pi/N$ are uniformly distributed over $[0,\pi)$ through the identities:

$$\sum_{k=1}^{N}\cos 2\theta_k = 0, \quad \text{and} \quad \sum_{k=1}^{N}\sin 2\theta_k = 0.$$

These are used to cancel out the linear factors

$$\sum_{k=1}^{N}\frac{\sigma^2}{2N}(\omega_1^2 - \omega_2^2)\cos 2\theta_k$$

and

$$\sum_{k=1}^{N}\frac{\sigma^2}{N}\omega_1\omega_2\sin 2\theta_k.$$

Finally, by noting that $(1 - x/m)^m$ converges to $\exp(-x)$ as m gets large, we arrive at (8.34) as the limiting case of (8.37).

To demonstrate that (8.34) holds in the $\mathbf{L}^2$ norm sense, it suffices to show the sequence of error functions

$$\mathcal{E}_N(\boldsymbol{\omega}) = \widehat{\beta}^N_{a(N)}(\boldsymbol{\omega}) - \exp(-\sigma^2 \|\boldsymbol{\omega}\|^2 / 2)$$

converge to zero in the above norm, that is, $\|\mathcal{E}_N\|_2 \longrightarrow 0$ as $N \longrightarrow \infty$. Now, since we have already demonstrated that $\mathcal{E}_N(\boldsymbol{\omega}) \longrightarrow 0$ pointwise, all we need to show in order to invoke the dominated convergence theorem is that the sequence $|\mathcal{E}_2(\boldsymbol{\omega})|, |\mathcal{E}_3(\boldsymbol{\omega})|, \ldots$ is uniformly bounded by a $\mathbf{L}^2$ function. Moreover, since

$$|\mathcal{E}_N(\boldsymbol{\omega})| \leq |\widehat{\beta}^N_{a(N)}(\boldsymbol{\omega})| + \exp(-\sigma^2 \|\boldsymbol{\omega}\|^2 / 2),$$

it suffices to show that each $|\widehat{\beta}^N_{a(N)}(\boldsymbol{\omega})|$ admits such a bound.

The main idea behind establishing such a bound is that the above mentioned sequence can be covered by a Gaussian in a neighborhood of the origin and by a function with sufficient decay at the tails, independent of N. Indeed, using the estimate $\mathrm{sinc}(u) \leq 1 - u^2/\pi^2$ for $|u| \leq \pi$, one can verify that

$$\begin{aligned}
\left|\widehat{\beta}^N_{a(N)}(\boldsymbol{\omega})\right| &= \prod_{k=1}^N \left|\mathrm{sinc}\left(\frac{\sqrt{6}\sigma}{\sqrt{N}} \boldsymbol{u}_{\theta_k}^T \boldsymbol{\omega}\right)\right| \\
&\leq \prod_{k=1}^N \left[1 - \frac{6\sigma^2 |\boldsymbol{u}_{\theta_k}^T \boldsymbol{\omega}|^2}{\pi^2 N}\right] \\
&\leq \exp\left(\boldsymbol{\omega}^T \mathbf{C}_1 \boldsymbol{\omega}\right) \qquad (\|\boldsymbol{\omega}\| < \delta)
\end{aligned}$$

where the matrix $\mathbf{C}_1$ and the number δ are independent of N. As far as the tail is concerned, the Cauchy-Schwarz inequality $|\boldsymbol{u}_{\theta_k}^T \boldsymbol{\omega}| \leq \|\boldsymbol{\omega}\|$, gives

$$\left|\widehat{\beta}^N_{a(N)}(\boldsymbol{\omega})\right| = \prod_{k=1}^N \left|\mathrm{sinc}\left(\frac{\sqrt{6}\sigma}{\sqrt{N}} \boldsymbol{u}_{\theta_k}^T \boldsymbol{\omega}\right)\right| \leq \frac{C_2}{\|\boldsymbol{\omega}\|^2} \qquad (\|\boldsymbol{\omega}\| \geq \delta).$$

where C_2 is again independent of N. Combining the above estimates, we see that

$$\left|\widehat{\beta}^N_{a(N)}(\boldsymbol{\omega})\right| \leq \exp\left(-\boldsymbol{\omega}^T \mathbf{C}_1 \boldsymbol{\omega}\right) + \frac{C_2}{\|\boldsymbol{\omega}\|^2}\left[1 - \mathrm{rect}\left(\frac{\|\boldsymbol{\omega}\|}{\delta}\right)\right]$$

for all $\boldsymbol{\omega}$. Since the function on the right is indeed in $\mathbf{L}^2$, this establishes the desired bound, and consequently, the norm convergence.

Appendix B

We begin with the observation that if $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ are symmetric (about the origin) and have a total mass of unity, then

$$\mathbf{C}_{f*g} = \mathbf{C}_f + \mathbf{C}_g,$$

where $\mathbf{C}_f$ denotes the covariance matrix of $f(\mathbf{x})$. To see this, note that $\hat{f}(0) = \hat{g}(0) = 1$ (unit mass), and that $\partial_i \hat{f}(0) = \partial_i \hat{g}(0) = 0$ (by symmetry). Using the multiplication-differentiation rule,

$$\int x_i x_j f(\mathbf{x}) \, d\mathbf{x} = -\partial_i \partial_j \hat{f}(0),$$

we then have

$$
\begin{aligned}
\mathbf{C}_{f*g}(i,j) &= \int x_i x_j (f*g)(\mathbf{x}) \, d\mathbf{x} \\
&= -\hat{g}(0)\partial_i \partial_j \hat{f}(0) - \hat{f}(0)\partial_i \partial_j \hat{g}(0) - \partial_i \hat{f}(0)\partial_j \hat{g}(0) - \partial_i \hat{g}(0)\partial_j \hat{f}(0) \\
&= -\partial_i \partial_j \hat{f}(0) - \partial_i \partial_j \hat{g}(0) \\
&= \mathbf{C}_f(i,j) + \mathbf{C}_g(i,j).
\end{aligned}
$$

Now, since the directional box distributions $\varphi_{a_k,\theta_k}(\mathbf{x})$ satisfy the above criteria,

$$\mathbf{C}_a^N = \sum_{k=1}^N \mathbf{C}_k$$

where $\mathbf{C}_k$ is the covariance matrix of $\varphi_{a_k,\theta_k}(\mathbf{x})$. We explicitly compute the component $\mathbf{C}(1,2)$; the remaining components can be similarly derived. Using the multiplication-differentiation rule again, we have

$$\mathbf{C}_k(1,2) = \int x_1 x_2 \varphi_{a_k,\theta_k}(\mathbf{x}) \, d\mathbf{x} = -\partial_1 \partial_2 \hat{\beta}_{a_k}(\mathbf{u}_{\theta_k}^T \boldsymbol{\omega})\Big|_{\boldsymbol{\omega}=0} = \frac{a_k^2}{24} \sin 2\theta_k.$$

Therefore,

$$\mathbf{C}_a^N(1,2) = \sum_k \mathbf{C}_k(1,2) = \frac{1}{24} \sum_k a_k^2 \sin 2\theta_k.$$

Finally, we note that $\mathbf{C}_a^N$ is positive-definite. This is simply because

$$\mathbf{u}^T \mathbf{C}_a^N \mathbf{u} = \int \mathbf{u}^T (\mathbf{x}\mathbf{x}^T)\mathbf{u} \, \beta_a^N(\mathbf{x}) \, d\mathbf{x} = \int (\mathbf{u}^T \mathbf{x})^2 \beta_a^N(\mathbf{x}) \, d\mathbf{x} > 0.$$

Alternatively, we also show that its eigenvalues

$$\lambda_{\max} = \frac{1}{2}\left(\sum a_k^2 + \sqrt{D}\right) \quad \text{and} \quad \lambda_{\min} = \frac{1}{2}\left(\sum a_k^2 - \sqrt{D}\right)$$

are strictly positive, where

$$D = \left(\sum a_k^2 \cos(2\theta_k)\right)^2 + \left(\sum a_k^2 \sin(2\theta_k)\right)^2.$$

This is obviously the case for $\lambda_{\max}$. Moreover,

$$\left(\sum_k a_k^2\right)^2 - D = \left(\sum_k a_k^2\right)^2 - \left(\sum_k a_k^2 \cos(2\theta_k)\right)^2 - \left(\sum_k a_k^2 \sin(2\theta_k)\right)^2$$

$$= 2\sum_{k \neq \ell} a_k^2 a_\ell^2 \left(1 - \cos(2\theta_k - 2\theta_\ell)\right) > 0.$$

This shows us that $\sum a_k^2 > \sqrt{D}$, so that $\lambda_{\min}$ is strictly positive as well.

Appendix C

Following definition (8.27), the dependence of orientation of the box spline $\beta_a^4(x)$ on the scale-vector can be expressed as

$$\tan \theta_a = v + \text{sign}(a_2 - a_4)\sqrt{1 + v^2} \qquad (0 < \theta_a < \pi) \qquad (8.38)$$

where

$$v = (a_3^2 - a_1^2)/(a_2^2 - a_4^2).$$

It can be verified that it is both necessary and sufficient that $a_2 > a_4$ (resp. $a_2 < a_4$) for the box spline to be oriented between $0 < \theta_a < \pi/2$ (resp. $\pi/2 < \theta_a < \pi$). Now note that the map $(a_1, a_2, a_3, a_4) \mapsto (v, \text{sign}(a_2 - a_4))$ uniquely determines the orientation of the box spline. This uniqueness is based on the argument that, for $0 < \theta_a < \pi/2$, (8.38) reduces to $\tan \theta_a = v + \sqrt{1 + v^2}$ since $a_2 > a_4$. This implicitly represents a one-to-one between θ_a and v over the domains $(0, \pi/2)$ and $(-\infty, \infty)$, since the map $\theta_a \mapsto \tan \theta_a$ from $(0, \pi/2)$ into $(0, \infty)$, and the map $v \mapsto v + \sqrt{1 + v^2}$ from $(-\infty, \infty)$ into $(0, \infty)$ are both strictly monotonic. Similarly, a one-to-one between θ_a and v over the domains $(\pi/2, \pi)$ and $(-\infty, \infty)$ can be established. In particular, we have

$$v = \frac{1}{2}(\tan \theta_a - \cot \theta_a)\,\text{sign}\left(\frac{\pi}{2} - \theta_a\right). \qquad (8.39)$$

Thus, given any orientation $\theta_a = \phi$, the corresponding v_ϕ is uniquely specified by (8.39). This establishes the first part of the proposition, since there trivially exists some positive vector $(a_1, \ldots, a_4)$ such that $(a_3^2 - a_1^2)/(a_2^2 - a_4^2) = v_\phi$.

As far as the bound is concerned, we observe that the elongation can be expressed as $\varrho_a = 1 + 2/(\gamma - 1)$, where $\gamma = \sum a_k^2 / \sqrt{D}_a \geq 1$. For a given orientation $\theta_a = \phi$, the components of the feasible scale-vectors bear the relation $(a_1^2 - a_3^2) = v_\phi(a_4^2 - a_2^2)$,

and thus we have that

$$\gamma = \frac{\sum a_k^2}{\sqrt{(a_3^2 - a_1^2)^2 + (a_2^2 - a_4^2)^2}}$$

$$= \frac{a_1^2 + a_3^2}{\sqrt{(a_3^2 - a_1^2)^2 + (a_2^2 - a_4^2)^2}} + \frac{a_2^2 + a_4^2}{\sqrt{(a_3^2 - a_1^2)^2 + (a_2^2 - a_4^2)^2}}$$

$$= \frac{1}{\sqrt{1 + v_\phi^2}} \frac{a_1^2 + a_3^2}{|a_1^2 - a_3^2|} + \frac{|v_\phi|}{\sqrt{1 + v_\phi^2}} \frac{a_2^2 + a_4^2}{|a_2^2 - a_4^2|} > \frac{1 + |v_\phi|}{\sqrt{1 + v_\phi^2}}$$

following the trivial inequalities $a_1^2 + a_3^2 > |a_1^2 - a_3^2|$, and $a_2^2 + a_4^2 > |a_2^2 - a_4^2|$. Consequently,

$$\varrho_a = 1 + \frac{2}{\gamma - 1} < \frac{1 + |v_\phi| + \sqrt{1 + v_\phi^2}}{1 + |v_\phi| - \sqrt{1 + v_\phi^2}}. \tag{8.40}$$

The above bound is tight since it can be approached arbitrary closely by making the scales a_ℓ and a_k ($\theta_\ell < \phi < \theta_k$) arbitrarily large.

Appendix D

Let $\mathbf{K}$ and $\mathbf{K}_\theta$ denote the kurtosis matrices of $f(\mathbf{x})$ and its rotation $f(\mathbf{R}_\theta^T \mathbf{x})$, respectively, where $\mathbf{R}_\theta$ is the rotation matrix. Observe that $\mathbf{L}_\theta$ and $\mathbf{L}$ are related as

$$\mathbf{L}_\theta = \int (\mathbf{x}\mathbf{x}^T)^2 f(\mathbf{R}_\theta^T \mathbf{x}) \, d\mathbf{x} = \int \mathbf{R}_\theta (\mathbf{y}\mathbf{y}^T)^2 \mathbf{R}_\theta^T f(\mathbf{y}) d\mathbf{y} \qquad (\mathbf{y} = \mathbf{R}_\theta^T \mathbf{x})$$

$$= \mathbf{R}_\theta \left(\int (\mathbf{y}\mathbf{y}^T)^2 f(\mathbf{y}) \, d\mathbf{y} \right) \mathbf{R}_\theta^T$$

$$= \mathbf{R}_\theta \mathbf{L} \mathbf{R}_\theta^T.$$

Similarly, we have $\mathbf{C}_\theta = \mathbf{R}_\theta \mathbf{C} \mathbf{R}_\theta^T$. Now since $\mathrm{tr}(\mathbf{A}\mathbf{B}) = \mathrm{tr}(\mathbf{B}\mathbf{A})$ and $\mathbf{R}_\theta^T \mathbf{R}_\theta = \mathbf{I}$,

$$\mathrm{tr}(\mathbf{C}_\theta) = \mathrm{tr}(\mathbf{R}_\theta \mathbf{C} \mathbf{R}_\theta^T) = \mathrm{tr}(\mathbf{C} \mathbf{R}_\theta^T \mathbf{R}_\theta) = \mathrm{tr}(\mathbf{C}).$$

Therefore,

$$\mathbf{K}_\theta = \mathbf{L}_\theta - \mathrm{tr}(\mathbf{C}_\theta)\mathbf{C}_\theta - 2\mathbf{C}_\theta^2 \mathbf{R}_\theta \mathbf{L} \mathbf{R}_\theta^T - \mathrm{tr}(\mathbf{C})\mathbf{R}_\theta \mathbf{C} \mathbf{R}_\theta^T - 2\mathbf{R}_\theta \mathbf{C}^2 \mathbf{R}_\theta^T$$

$$= \mathbf{R}_\theta(\mathbf{L} - \mathrm{tr}(\mathbf{C})\mathbf{C} - 2\mathbf{C}^2)\mathbf{R}_\theta^T = \mathbf{R}_\theta \mathbf{K} \mathbf{R}_\theta^T.$$

Our claim follows immediately, since

$$\|\mathbf{K}_\theta\|^2 = \mathrm{tr}(\mathbf{K}_\theta^T \mathbf{K}_\theta) = \mathrm{tr}(\mathbf{R}_\theta \mathbf{K}^T \mathbf{K} \mathbf{R}_\theta^T) = \mathrm{tr}(\mathbf{K}^T \mathbf{K}) = \|\mathbf{K}\|^2.$$

Appendix E

In order to compute the kurtosis matrix, we only need to evaluate the fourth-order moments; the second-order moments are alrea dy known. In particular, using Fourier identities similar to the ones used in Appendix B, one can derive the following expressions:

$$\int x_1^4 \beta_a(x)\, dx = \frac{1}{4}\mu_4 \left(4a_1^4 + a_2^4 + a_4^4\right) + \frac{1}{2}\mu_2^2 \left(6a_1^2 a_2^2 + 6a_1^2 a_4^2 + 3a_2^2 a_4^2\right),$$

$$\int x_1^3 x_2 \beta_a(x)\, dx = \frac{1}{4}\mu_4 \left(a_2^4 - a_4^4\right) + \frac{3}{2}\mu_2^2 a_1^2 \left(a_2^2 - a_4^2\right),$$

$$\int x_1^2 x_2^2 \beta_a(x)\, dx = \frac{1}{4}\mu_4 \left(a_2^4 + a_4^4\right) + \frac{1}{2}\mu_2^2 \left(a_1^2 a_2^2 + a_1^2 a_4^2 + a_2^2 a_3^2 + a_3^2 a_4^2 - a_2^2 a_4^2 + 2a_1^2 a_3^2\right),$$

$$\int x_1 x_2^3 \beta_a(x)\, dx = \frac{1}{4}\mu_4 \left(a_2^4 - a_4^4\right) + \frac{3}{2}\mu_2^2 a_3^2 \left(a_2^2 - a_4^2\right),$$

$$\int x_2^4 \beta_a(x)\, dx = \frac{1}{4}\mu_4 \left(4a_3^4 + a_2^4 + a_4^4\right) + \frac{1}{2}\mu_2^2 \left(6a_2^2 a_3^2 + 6a_3^2 a_4^2 + 3a_2^2 a_4^2\right),$$

where $\mu_4 = 1/80$ and $\mu_2 = 1/12$ denote the fourth and second-order moments of $\mathrm{Box}_1(x)$, respectively. These provide the components of the matrix $\mathbf{L}_a$, which in turn leads to the following simple expression for the kurtosis matrix

$$\mathbf{K}_a = \mathbf{L}_a - \mathrm{tr}(\mathbf{C}_a)\mathbf{C}_a - 2\mathbf{C}_a^2 = (\mu_4 - 3\mu_2^2) \begin{pmatrix} a_1^4 + \frac{1}{2}(a_2^4 + a_4^4) & \frac{1}{2}(a_2^4 - a_4^4) \\ \frac{1}{2}(a_2^4 - a_4^4) & a_3^4 + \frac{1}{2}(a_2^4 + a_4^4) \end{pmatrix}. \tag{8.41}$$

Finally, from (8.41), we get $\|\mathbf{K}_a\|^2 = \sum_{k=1}^4 a_k^8 + (a_1^4 + a_3^4)(a_2^4 + a_4^4)$.

We note that the negative factor $(\mu_4 - 3\mu_2^2)$ in (8.41) is in fact the kurtosis of $\mathrm{rect}(x)$, the sub-Gaussian constituent of the box spline. The fact that $\mathbf{K}_a$ is negative-definite is thus consistent with the sub-Gaussian nature of the resulting box spline.

Appendix F

Given a discrete function $c[n]$ and a point x on $\mathbf{R}^2$, we outline a technique for the fast evaluation of the sum

$$\sum_{n \in \mathbf{Z}^2} c[n]\beta_{\mathrm{ZP}}(n - x) \tag{8.42}$$

A sketch of the partitions of the piecewise polynomial $\beta_{\mathrm{ZP}}(x)$ is provided in Figure 8.10. The exact functional forms of the ZP box spline $2\beta_{\mathrm{ZP}}(x)$ corresponding to these partitions can be found in [92]. Since $\beta_{\mathrm{ZP}}(x)$ has a compact support, this is in fact a finite sum, and requires at most seven evaluations of the function $\beta_{\mathrm{ZP}}(\cdot - x)$ for any arbitrary translation x. This is illustrated in Figure 8.10, where the red dots
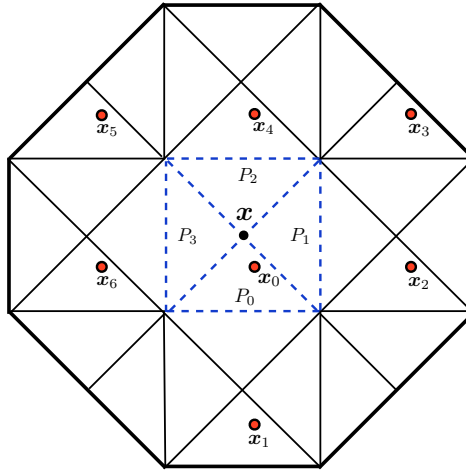
Figure 8.10: The translated box spline $\beta_{ZP}(\cdot - x)$. The red dots $x_1, \ldots, x_7$ correspond to the points on the Cartesian lattice, and the triangular regions $P_1, \ldots, P_4$ are different partitions of the ZP, which together constitute a unit cell of the lattice.

$x_0, \ldots, x_6$ denote the lattice points that intersect the support of $\beta_{ZP}(\cdot - x)$. Thus, one needs to evaluate the translated ZP at the points $x_0, \ldots, x_6$ in order to compute the sum. The drawback here is that naive evaluation of the spline at every x_j requires a decision-making to figure out the associated partition before computing the corresponding polynomial.

The redundancy that we exploit is as follows: Consider the triangular regions $P_0, \ldots, P_3$ marked with blue dashed lines in Figure 8.10 corresponding to the four different partitions of the ZP. These together constitute a unit cell of the lattice, and hence only one lattice point intersects them. The figure shows a particular instance where this point, denoted by x_0, lies in P_0. This clearly fixes the partitions of the remaining lattice points $x_1, \ldots, x_6$. Thus, if we denote the polynomials corresponding to these partitions by $\rho_{0,0}(x), \ldots, \rho_{0,6}(x)$, then the sum in (8.42) is simply given by $\sum_{j=0}^{6} c[x_j]\rho_{0,j}(x_j)$. More generally, if x_0 intersects the partition P_i ($0 \leq i \leq 3$), and if we denote the corresponding polynomials by $\rho_{i,j}(x)$, then the sum is given by $\sum_{j=0}^{6} c[x_j]\rho_{i,j}(x_j)$. Thus, we have the computational advantage that at most two binary decisions are required to determine the ZP partitions corresponding to the points x_j, and that we can pre-compute the coefficients of the polynomials $\rho_{i,j}(x)$.

Appendix G

The following is the probabilistic interpretation of the Central Limit Theorem (CLT) used for approximating the Gaussian distribution using box splines. Given that all forms of the CLT are deep results, we wonder if there are some interesting applications of this idea in the stochastic setting as well?

Let X be random vector on the plane, whose realizations are distributed on a line passing through the origin (e.g., one of the coordinate axes). Thus, X is completely specified by a probability measure $\mu(t)$ on the real line. Suppose that

$$\int t \, d\mu(t) = 0, \quad \text{and} \quad \int t^2 \, d\mu(t) = 1.$$

Finally, given $0 \leq \theta < \pi$, let us denote the rotation matrix on the plane by

$$\mathcal{R}_\theta = \left(\begin{array}{cc} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{array} \right).$$

Then the following holds.

Theorem 30 (Radial CLT). *For any integer N, fix the rotations $\theta_k = (k-1)\pi/N$ for $1 \leq k \leq N$. Suppose that $X_1, X_2, \ldots, X_N$ are independent and identically distributed copies of X. Then the random vector*

$$Z_N = \frac{1}{\sqrt{N}} \left(\mathcal{R}_{\theta_1} X_1 + \mathcal{R}_{\theta_2} X_2 + \cdots + \mathcal{R}_{\theta_N} X_N \right)$$

converges to the standard normal distribution on the plane as N gets large. More precisely, for any Borel subset B on the plane,

$$\lim_{N \longrightarrow \infty} \mathrm{Prob}(Z_N \in B) = \frac{1}{2\pi} \int_B \exp\left(-\frac{\|x\|^2}{2} \right) \, dx.$$

The above theorem suggests a means of picking vectors from the normal distribution on the plane. Apparently, this method is quite efficient, and works well even with a small number of rotations (see Figure 8.11). The following is a MATLAB simulation for the case, where X is $\sqrt{12}$ times the uniform distribution over $(-1/2, 1/2)$ on the x-coordinate):

```
%—————————
N=10; draws=10000;
theta=(0:N-1)*pi/N;

Z1=zeros(draws,1);
```

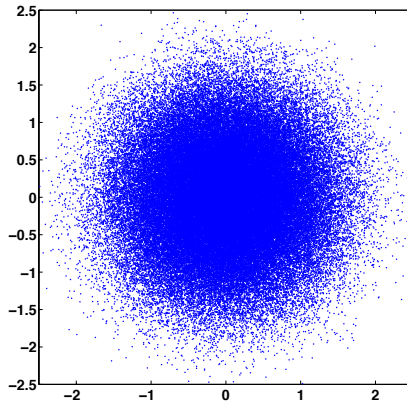**Figure 8.11:** The scatter plot of Z_N over $10,000$ realizations, with $N = 10$ rotations.

```
Z2=zeros(draws,1);

for n=1:draws          % outer loop (realizations)
X1=0;
X2=0;
r=sqrt(12)*(rand(N,1)-0.5*ones(N,1));
for k=1:N              % inner loop (rotations)
X1=X1+r(k,1)*cos(theta(1,k));
X2=X2+r(k,1)*sin(theta(1,k));
end

Z1(n,1)=sqrt(1/N)*X1;

Z2(n,1)=sqrt(1/N)*X2;
end

figure(2),scatter (Z1,Z2); axis image
%——————————-
```

Chapter 9

Smoothing using space-variant filters

Abstract — We propose two smoothing filters based on non-linear, space-variant averaging. The first of these is modeled on anisotropic (Gaussian) diffusion. To approximate the anisotropic Gaussian, we use the four-directional box spline introduced in Chapter 8. The space-variance in this case is in terms of the size, elongation, and orientation of the box splines.

The other scheme is based on a space-variant form of the Gaussian bilateral filter, which involves a range filter along with spatial filter. The adaptability in this case is in terms of the size of the spatial filter. We develop an efficient constant-time algorithm for implementing the space-variant filter. This is achieved by approximating the spatial Gaussian filter using four-directional box splines, and by (locally) approximating the Gaussian range filter using trigonometric functions.

To demonstrate the working of these filters, we perform some denoising experiments[1].

Smoothing of images using Gaussian filters (isotropic diffusion) results in excessive blurring of image features. The more sophisticated anisotropic diffusion is known to perform better in such cases [31, 84]. To overcome the shortcomings of local smoothing, researchers have also proposed various global and multiscale approaches, where, e.g., one proceeds by minimizing a global energy functional such as the total-variation functional [94], or by exploiting the sparsity of the wavelet representation [23]. We refer the interested reader to [95] for an exhaustive review and comparison

---

[1]This chapter is based on [77]: K. N. Chaudhury, A. Muñoz-Barrutia, M. Unser, "Fast space-variant elliptical filtering using box splines," IEEE Transactions on Image Processing, vol. 19, no. 9, pp. 2290-2306, 2010; and on the article, K. N. Chaudhury, D. Sage, M. Unser, "Fast $O(1)$ bilateral filtering using trigonometric range kernels", submitted to IEEE Transaction on Image Processing.

of many such methods. The focus of the present chapter is, however, solely on local smoothing. Similar to anisotropic diffusion, we propose the use of certain smoothing mechanisms which are based on non-linear averaging. We develop fast and efficient algorithms for implementing these schemes.

9.1 Space-variant anisotropic smoothing

In our first scheme, we propose to smooth the image using a space-variant form of the anisotropic Gaussian filter, where the anisotropy of the filter is changed from point-to-point inside the image. We approximate the anisotropic Gaussian using the four-directional box spline introduced in Chapter 8. The main idea is to adapt the size, elongation, and orientation of the filter to the local image features. In effect, we locally average the corrupted image using box splines that have been elongated along the image feature (orthogonal to the local gradient). This induces more smoothing along the features, and less across it. As a result, we are able to suppress the ambient noise, while preserving the sharpness of the image features at the same time.

9.1.1   Anisotropy estimation

To derive an estimate of the local image anisotropy, we use the so-called *structure tensor* [96]. The local orientation in this case is estimated through the minimization of a weighted-norm of the directional derivative. Let $D_\theta f(x)$ be the directional derivative of $f(x)$ along $u_\theta = (\cos\theta, \sin\theta)$. Consider a suitable averaging window $w(x)$ with support Ω. Then the estimate of the local orientation is given

$$\theta^\star(x) = \arg \min_{0 \leq \theta < \pi} \int_\Omega w(y) |D_\theta f(x-y)|^2 \, dy. \qquad (9.1)$$

Note that the objective in (9.1) can be expressed as a quadratic form. Indeed, by expressing the directional derivative in terms of the gradient $g(x)$,

$$D_\theta f(x) = u_\theta^T g(x),$$

we can write the objective as

$$u_\theta^T J(x) u_\theta \qquad (9.2)$$

where

$$J(x) = \int_\Omega w(y) (gg^T)(x-y) \, dy.$$

The 2×2 matrix J is called the *structure tensor*. Note that J is symmetric and positive semidefinite. Following its eigen decomposition, it is immediate that $\theta^\star(x)$ is simply the direction of the eigenvector corresponding to the minimum eigenvalue of J.

Using the fact that both the eigenvalues of J are non-negative, we propose to estimate the elongation as follows. Let the eigenvalues be $0 \leq \lambda_{\min} \leq \lambda_{\max}$. We set $\varrho^\star =$

$\lambda_{\max}/\lambda_{\min}$ if both eigenvalues are non-zero, equal to 1 if both are zero (locally isotropic intensity), and equal to $\max(1, \lambda)$ if only one of the eigenvalues λ is non-zero. On the other hand, we estimate the size of the box spline as $s^\star = \lambda_{\max} + \lambda_{\min}$.

We note that the structure tensor matrices can be efficiently computed using simple convolution and pointwise operations. We refer the reader to [96, Chapter 13] for implementation details.

9.1.2 Smoothing algorithm

The anisotropy estimates $s^\star, \varrho^\star$, and $\theta^\star$ are used to compute the scale-vector of the four-directional box spline. This is done using the algorithm described in §6.2 of Chapter 8. The main steps of the our denoising algorithm are as follows.

- Computation of the structure-tensor.

- Computation of the estimates $s^\star, \varrho^\star$, and $\theta^\star$ at every location using the eigen decomposition of the structure tensor. This is used to determine the scale-vector $a(n)$ of the Gaussian-like box spline.

  Isotropic box splines are used in the uniform-intensity regions; we set $a = (\sigma, \sigma, \sigma, \sigma)$, where σ is proportional to the noise variance.

- Pre-integration of the corrupted image using running-sums.

- Computation of the parameters of the finite-difference mesh using $a(n)$, and its application to the pre-integrated image.

9.1.3 Denoising experiments

We now apply our algorithm for denoising synthetic and real images corrupted with additive Gaussian noise. We compare the results obtained using our algorithm with those obtained using the isotropic Gaussian filter and the Perona-Malik diffusion filter [31]. The algorithm was implemented in Java on a Macintosh 2.8 GHz Intel dual-core system. The average execution time of our algorithm was 600 milliseconds for a 512×512 image. This included the time for computing the structure-tensor, the running-sums, the optimal scale-vector, the interpolated samples, and the finite-differences.

We use the standard test image of *Barbara* and corrupt it with white Gaussian noise. We use the variance of the noise to set the size of the Gaussian used for isotropic smoothing. The parameters used for the Perona-Malik filter are typical: time step of 0.1, conductance in the range of $10 \sim 30$, and a total of $15 \sim 30$ iterations. The parameters were manually tuned to optimize the PSNR, and also to avoid blocking artifacts. Figure 9.1 shows the results obtained from the different filters. As far as the quantitative evaluation of the filters is concerned, our algorithm clearly
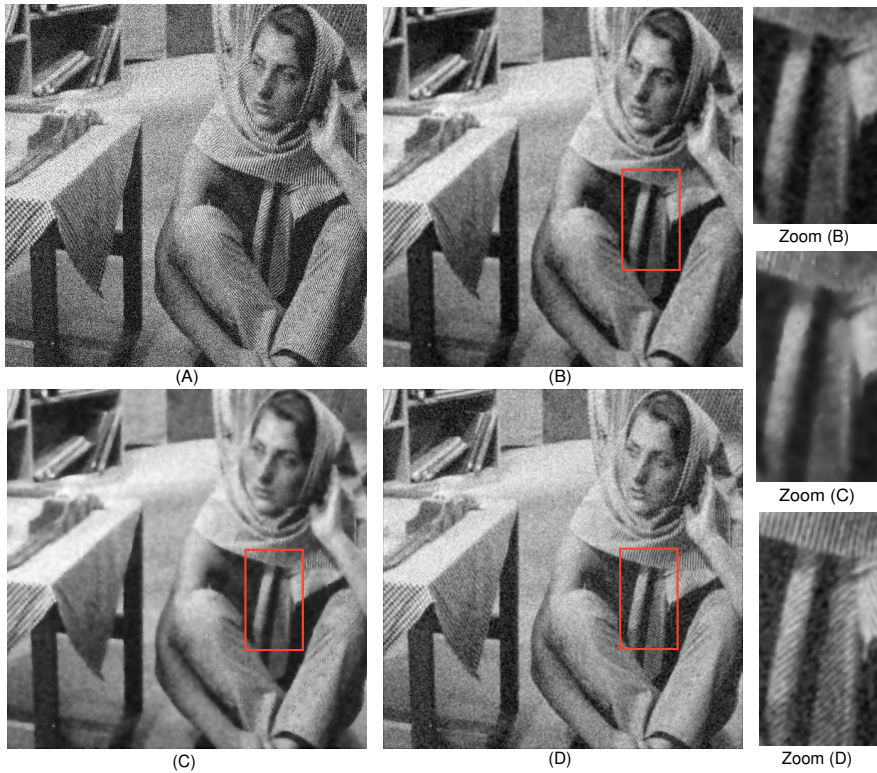
Figure 9.1: Denoising results on a test image. (**A**) *Barbara* corrupted with additive Gaussian noise, PSNR = 18.0 dB; (**B**) Isotropic smoothing, PSNR = 23.10 dB; (**C**) Diffusion filtering, PSNR = 23.25 dB; (**D**) Our algorithm, PSNR = 23.58 dB. The algorithms were tuned for the best PSNR.
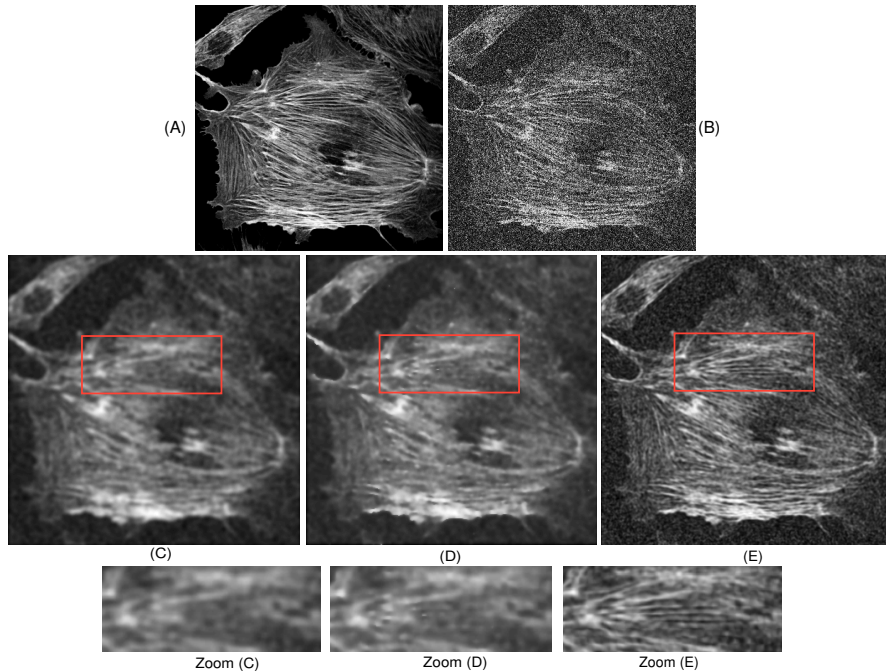
Figure 9.2: Denoising results on a real biological image. (**A**) Noise-free immunofluorescence image of actin fibres (Courtesy of C. Aemisegger, CMIA, University of Zürich); (**B**) Image corrupted with additive Gaussian noise, PSNR = 12.20 dB; (**C**) Isotropic smoothing, PSNR = 15.38 dB; (**D**) Diffusion filtering, PSNR = 15.50 dB; (**E**) Our algorithm, PSNR = 15.80 dB.

outperforms both isotropic and diffusion filters in terms of the Peak-Signal-to-Noise-Ratio (PSNR); see Table 9.1. Moreover, as shown in the zoomed-in sections of the denoised images, the oriented stripes on the clothes of Barbara are quite faithfully restored by our algorithm. A significant amount of blurring of the stripes is seen in the results obtained using isotropic and diffusion filters. The non-linear diffusion filter, however, tends to perform better at low PSNRs in the range of 5-10 dB.

We next compare the results on the real biological image shown in Figure 9.2. This particular fluorescence image exhibits numerous elongated fiber-like structures. We use a lower PSNR of around 12 dB. The parameters of the isotropic filter and the diffusion filter are set as in the previous case, except that the iteration count for the latter is increased to 15. As before, the improvement of the PSNR obtained using our filter is higher. Importantly, as again apparent from the image zooms, our algorithm causes significantly less merging of the close fibers and blurring of the finer ones.

Table 9.1: Comparison of the PSNRs of the filters at different noise levels using the test image of *Barbara*.

Input PSNR (dB)	10.0	12.0	14.0	16.0	18.0	20.0
Isotropic filter	15.38	18.20	20.20	21.65	23.10	24.30
Diffusion filter	**15.48**	18.31	20.30	21.70	23.25	24.35
Our filter	15.45	**18.38**	**20.57**	**21.94**	**23.58**	**24.56**

9.2   Space-variant bilateral filter

The denoising strategy discussed in the previous section was based on local spatial filtering. This takes advantage of the similarity of pixels that are spatially close. The hypothesis used here is that the "neighborhood" pixels tend to be more correlated to the pixel of interest than the distant pixels. As a result, we were able to considerably reduce the noise by simply averaging the neighborhood pixels. The hypothesis, however, breaks down in the vicinity of sharp transitions, e.g., in the vicinity of an edge. The pixels one side of the edge tend to be better correlated than those on the other side of the edge. Isotropic averaging in such cases results in undue blurring of image features. This led us to consider space-variant anisotropic averaging, where we controlled the shape of the averaging neighborhood at every pixel based on the local image characteristics.

An alternative approach is that of *bilateral filtering* proposed by Tomasi and Manduchi [32]. Along with a spatial filter, the bilateral filter also involves a range filter. The role of the range filter is to penalize large deviations in intensities, typically between pixels lying on different sides of an edge. As a result, the averaging is biased towards pixels which are close both in space and photometric intensity to the pixel of interest. Let the choice of spatial and range filter be $w(x)$ and $g(s)$, respectively. The output of the bilateral filter is given by

$$\tilde{f}(x) = \eta^{-1} \int w(y) g\big(|f(x) - f(x-y)|\big) f(x-y) \, dy$$

where η is the normalizing factor:

$$\eta = \int w(y) g\big(|f(x) - f(x-y)|\big) \, dy.$$

The bilateral filter was originally introduced by Tomasi and Manduchi [32] as a edge-preserving operator. Subsequently, this simple and non-iterative operator has found widespread use in several image processing, computer graphics, and computer vision applications. This includes denoising [97, 98, 99, 100], texture editing [101], demosaicing [102], optical-flow estimation [103, 104], and stereo matching [105, 106], to name a few.

The most commonly used bilateral filter is one where $g(s)$ is a 1-dimensional Gaussian and $w(\boldsymbol{x})$ is an isotropic Gaussian. We call this the *Gaussian bilateral filter*. In this Section, we propose a *space-variant* form of the Gaussian bilateral filter. The idea is to vary the width of $w(\boldsymbol{x})$ based on the local structural information—we use wide Gaussians in the homogenous regions, and narrow ones in the vicinity of edges. The range filter is kept fixed.

Our goal is to implement the space-variant bilateral filter in constant time (irrespective of the size of the spatial filter), without any noticeable degradation of filter response. It was shown in Chapter 8 that the four-directional box splines provide a close approximation of isotropic Gaussians. We use these box splines to approximate the Gaussian spatial filter $w(\boldsymbol{x})$. On the other hand, the fixed range filter $g(s)$ is approximated by an appropriate polynomial. The idea of accelerating the implementation of the standard bilateral filter using a polynomial range filter was first proposed in [107]. This constant-time implementation was later generalized to include arbitrary spatial and range filters by several researchers; e.g., see [108] and the references therein.

9.2.1 Constant-time bilateral filtering using shiftable range kernels

Let $\beta_a(\boldsymbol{x})$ be the four-directional box spline of equal scale a along the four directions. Let $a(\boldsymbol{x})$ be the scale used at position $\boldsymbol{x}$. The output of the bilateral filter at location $\boldsymbol{x}$ is given by

$$\tilde{f}(\boldsymbol{x}) = \eta^{-1} \int \beta_{a(\boldsymbol{x})}(\boldsymbol{y}) g\big(|f(\boldsymbol{x}) - f(\boldsymbol{x}-\boldsymbol{y})|\big) f(\boldsymbol{x}-\boldsymbol{y}) \, d\boldsymbol{y} \qquad (9.3)$$

where

$$\eta = \int \beta_{a(\boldsymbol{x})}(\boldsymbol{y}) g\big(|f(\boldsymbol{x}) - f(\boldsymbol{x}-\boldsymbol{y})|\big) \, d\boldsymbol{y}. \qquad (9.4)$$

We assume that the range filter $g(s)$ is symmetric, so that we can replace $g(|s|)$ by $g(s)$ in the above formula. We will require this symmetry in the sequel.

In keeping with the notation of Chapter 8, we denote the spatial average obtained using the isotropic box spline as

$$\overline{f}(\boldsymbol{x}) = \int \beta_{a(\boldsymbol{x})}(\boldsymbol{y}) f(\boldsymbol{x}-\boldsymbol{y}) \, d\boldsymbol{y}. \qquad (9.5)$$

It is implied that the scale a depends on the position $\boldsymbol{x}$. The main point is that we can compute this spatial average in constant-time by exploiting the overlaps involved in the summation, and the particular structure of the box spline window.

Note that, unlike the transformation $f(\cdot) \mapsto \overline{f}(\cdot)$, the transformation $f(\cdot) \mapsto \tilde{f}(\cdot)$ is not linear. This is due to the presence of the range filter $g(s)$ in (9.3). Our goal is to

express (9.3) in terms of (9.5). This will allow us to compute the bilateral filter in constant-time.

We show that this can be done if the range filter $g(s)$ is symmetric and *shiftable*. We call a class of functions $\mathbf{G}$ as being shiftable if there exists a (fixed) set of functions $\varphi_1(s),\dots,\varphi_N(s)$ such that, for all $g(s)$ belonging to $\mathbf{G}$,

$$g(s-\tau)=\sum_{i=1}^{N}a_i(\tau)\varphi_i(s)\qquad(\tau\in\mathbf{R}).$$

The functions $\varphi_i(x)$ are called the basis functions. They do not depend on the translation parameter τ. It is the coefficients $a_i=a_i(\tau)$ that capture the action of the translation. Concrete instances of $\mathbf{G}$ are the class of polynomials $\mathbf{P}_N$ of degree at most N, and the class of trigonometric polynomials $\mathbf{T}_N$ of degree at most N. To see this, note that every $g(s)$ belonging to $\mathbf{P}_N$ is spanned by the basis functions

$$\varphi_1(s)=1,\varphi_2(s)=s,\dots,\varphi_N(s)=s^N,$$

and that every translate of every $\varphi_i(s)$ can be expressed in terms of $\varphi_1(s),\varphi_2(s),\dots,\varphi_i(s)$. On the other hand, every $g(s)$ belonging to the class $\mathbf{T}_N$ is spanned by the basis functions

$$\varphi_1(s)=1,\varphi_2(s)=e^{jns},\dots,\varphi_N(s)=e^{jNs}.$$

To see why $g(s)$ is shiftable, simply note that $\varphi_i(s-\tau)=e^{-jn\tau}\varphi_i(s)$.

Let $g(s)$ be shiftable and symmetric. Then for some basis functions $\varphi_1(s),\dots,\varphi_N(s)$ we can write

$$g(s-\tau)=\sum_{i=1}^{N}a_i(\tau)\varphi_i(s).$$

Therefore,

$$g(|f(\mathbf{x})-f(\mathbf{x}-\mathbf{y})|)=g(f(\mathbf{x}-\mathbf{y})-f(\mathbf{x}))=\sum_{i=1}^{N}a_i(\mathbf{x})\varphi_i(f(\mathbf{x}-\mathbf{y}))\qquad(9.6)$$

where we use the shorthand $a_i(\mathbf{x})=a_i(f(\mathbf{x}))$. Plugging (9.6) into (9.3), we get

$$\tilde{f}(\mathbf{x})=\eta^{-1}\sum_{i=1}^{N}a_i(\mathbf{x})\int\beta_{a(\mathbf{x})}(\mathbf{y})f(\mathbf{x}-\mathbf{y})\varphi_i(f(\mathbf{x}-\mathbf{y}))\,d\mathbf{y}$$

where

$$\eta=\sum_{i=1}^{N}a_i(\mathbf{x})\int\beta_{a(\mathbf{x})}(\mathbf{y})\varphi_i(f(\mathbf{x}-\mathbf{y}))\,d\mathbf{y}.$$

Let us define the *auxiliary images*

$$\phi_i(\mathbf{x})=f(\mathbf{x})\varphi_i(f(\mathbf{x}))\quad\text{and}\quad\psi_i(\mathbf{x})=\varphi_i(f(\mathbf{x})).$$

We can then write (9.3) as

$$\tilde{f}(\boldsymbol{x}) = \frac{\sum_i a_i(\boldsymbol{x})\overline{\phi_i}(\boldsymbol{x})}{\sum_i a_i(\boldsymbol{x})\overline{\psi_i}(\boldsymbol{x})}. \tag{9.7}$$

The gives us the following $O(1)$ algorithm for the bilateral filter: We first set up the auxiliary images and the coefficients $d_n(\boldsymbol{x})$ from the input image. We then average each of the auxiliary images using a $O(1)$ algorithm (this can be done in parallel). The samples of the filtered image is then given by the simple sum and division. In particular, for an image of size $M \times M$, we can compute the spatial averages for any arbitrary $w(\boldsymbol{x})$ at the cost of $O(M^2 \log_2 M)$ operations using the Fourier transform. As mentioned earlier, this can further be reduced to a total of $O(M^2)$ operations using specialized spatial kernels [28, 109, 77].

Two concrete instances of shiftable functions are the class of polynomials, and the class of trigonometric functions. The generators for the former are precisely the monomials

$$\varphi_1(s) = 1, \ \varphi_2(s) = s, \ \cdots, \ \varphi_N(s) = s^N.$$

On the other hand, the generators for the class of trigonometric functions of frequency at most N are the complex exponentials:

$$\varphi_1(\theta) = 1, \ \varphi_2(\theta) = \exp(j\theta), \ \cdots, \ \varphi_N(\theta) = \exp(jN\theta).$$

The idea of using polynomials (more precisely, Taylor polynomials) to approximate the Gaussian range kernel was proposed by Porikli in [107]. As will be shown in the sequel, this has some serious drawbacks. We show how these can be fixed using instead trigonometric functions.

9.2.2 Trigonometric range kernel

Note that $g(s)$ must have some additional properties to qualify as a valid range kernel (besides being symmetric). Namely, $g(s)$ must be non-negative, and must be monotonic in that $g(s_1) \leq g(s_2)$ whenever $|s_1| > |s_2|$. In particular, it must have a peak at the origin. This ensures that large differences in intensity get more penalized than small differences, and that the bilateral filter behaves purely as a spatial filter in a region having uniform intensity. Moreover, one must also have some control on the variance (effective width) of $g(s)$. We now address these design problems in order.

The properties of symmetry, non-negativity, and monotonicity are simultaneously enjoyed by the family of *raised cosines* of the form

$$g(s) = \big[\cos(\gamma s)\big]^N \qquad (-T \leq s \leq T).$$

Writing $\cos\theta = (e^{j\theta} + e^{-j\theta})/2$, and applying the binomial theorem, we see that

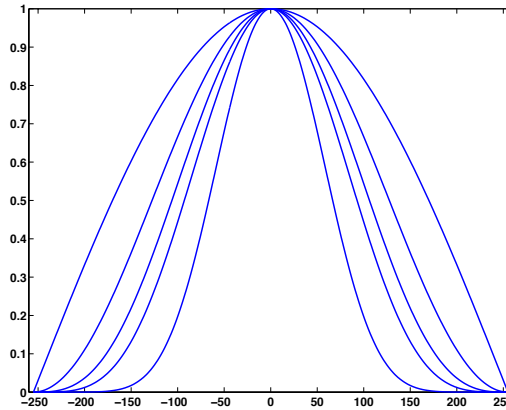$$g(s) = \sum_{n=0}^{N} 2^{-N}\binom{N}{n}\exp\big(j(2n-N)\gamma s\big).$$

Figure 9.3: The family of raised cosines $g(s) = (\cos \theta)^N$ over the dynamic range $-255 \leq s \leq 255$, where $\theta = 0.5\pi s/255$. The powers (outer to inner curves) are respectively $N = 1, 2, 3, 4$, and 6. Note that the functions become more Gaussian-like as their degree increases. We show that they indeed converge to a Gaussian (after appropriate normalization) as N gets large.

Since $g(s)$ has a total of $(N+1)$ terms, this gives a total of $2(N+1)$ auxiliary images. The central term $n = N/2$ is constant when N is even, and we have one less auxiliary image to process in this case.

Figure 9.3 shows the raised cosines of degree $N = 1$ to $N = 5$. It is seen that the functions become more Gaussian-like as the degree increases. Indeed, a remarkable property of the raised cosines is that they provide very close approximates of the Gaussian. More precisely, it can be shown that (cf. Appendix A for a proof):

$$\lim_{N \longrightarrow \infty} \left[\cos \left(\frac{\gamma s}{\sqrt{N}} \right) \right]^N = \exp \left(-\frac{\gamma^2 s^2}{2} \right). \tag{9.8}$$

The crucial fact is that the rate of convergence is much faster than that of the Taylor polynomials, which were used to approximate the Gaussian range kernel in [107]. In particular, we can obtain a approximation comparable to that achieved using polynomials using fewer number of terms. This is important from the practical standpoint. In Figure 9.4, we consider the target Gaussian kernel $\exp(-s^2/2\sigma^2)$, where $\sigma = 80$. We approximate this using the raised cosine of degree 4, which has 3 terms. We also plot the polynomial corresponding to the 3-term Taylor expansion of the Gaussian, which is used in for approximating the Gaussian in [107]. It is clear that the approximation quality of the raised cosine is superior to that offered by a Taylor polynomial having equal number of terms. In particular, note that the Taylor approximation does not automatically offer the crucial monotonic property.

The approximation in (9.8) also suggests a means of controlling the variance of the raised cosine, namely, by controlling the variance of the target Gaussian. The target
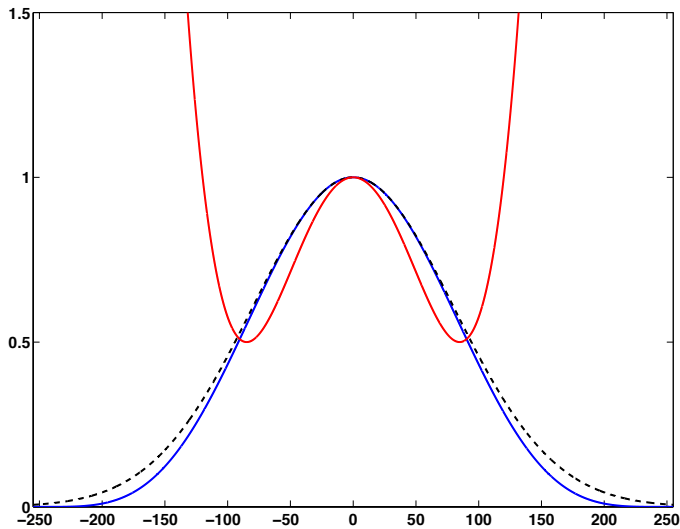
Figure 9.4: Approximation of the Gaussian $\exp(-x^2/2\sigma^2)$ (dashed black curve) over the interval $[-255, 255]$ using the Taylor polynomial (solid red curve) and the raised cosine (solid blue curve). We set $\sigma = 80$, and use $N = 4$ for the raised cosine in (9.8). The raised cosine is of the form $a_0 + a_1 \cos(2\theta) + a_2 \cos(4\theta)$ in this case. We use a 3-term Taylor polynomial of the form $b_0 + b_1 x^2 + b_2 x^4$. It is clear that the raised cosine offers a much better approximation than its polynomial counterpart. In particular, note how the polynomial blows up beyond $|x| > 100$.

(a)

(b)

(c)

(d)
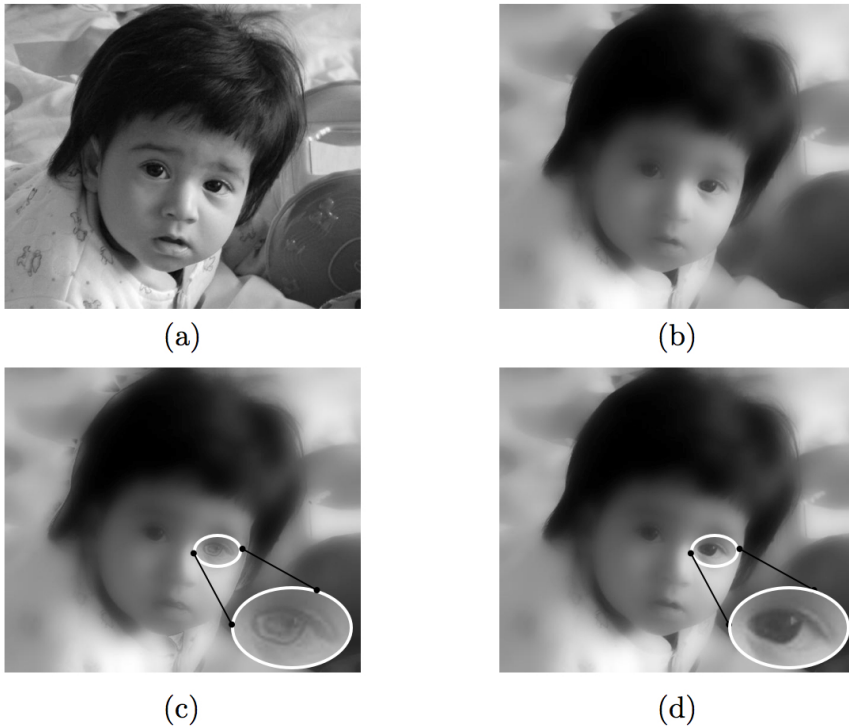
Figure 9.5: Comparison of the implementations of the Gaussian bilateral filter on the grayscale image *Isha* (size 600×512). The filter settings are $\sigma_s = 15$ and $\sigma_r = 80$. (a) The original image; (b) Direct implementation of the bilateral filter; (c) Output obtained using polynomial kernel [107]; and (d) The output of our algorithm. Note the strange artifacts in (c), particularly around one of the eyes (zoomed insets). This is on account of the form of the polynomial approximation shown in Figure 9.4. The standard deviation of the error between (b) and (c) is 6.5, while that between (b) and (d) is 1.2.

Gaussian (with normalization) has a fixed variance of γ^{-2}. This can be increased simply by rescaling the argument of the cosine in (9.8) by some $\rho > 1$. In particular, for sufficiently large N,

$$\left[\cos\left(\frac{\gamma s}{\rho\sqrt{N}}\right)\right]^{N} \approx \exp\left(-\frac{s^2}{2\rho^2\gamma^{-2}}\right). \tag{9.9}$$

The variance of the target Gaussian (again with normalization) has now increased to $\rho^2\gamma^{-2}$. A fairly accurate estimate of the variance of the raised cosine is therefore $\sigma^2 \approx \rho^2\gamma^{-2}$. In particular, we can increase the variance simply by setting $\rho = \gamma\sigma$ for all $\sigma > \gamma^{-2}$, provided N is large enough.

Table 9.2: N_0 is the minimum degree of the raised cosine required to approximate a Gaussian of standard deviation σ on the interval $[-255, 255]$. The a prior estimate $\lceil(\gamma\sigma)^{-2}\rceil$ is also shown.

σ	200	150	100	80	60	50	40
N_0	1	2	3	4	5	7	9
$\lceil(\gamma\sigma)^{-2}\rceil$	1	2	3	5	8	17	11

Bringing down the variance below γ^{-2}, on the other hand, is more subtle. This cannot be achieved simply by rescaling with $\rho < 1$ on account of the oscillatory nature of the cosine. For instance, setting $\rho < 1$ can cause $g(s)$ to become non-negative, or loose its monotonicity. The only way of doing so is by increasing the degree of the cosine (cf. Figure 9.3). In particular, N must be large enough so that the argument of $\cos(\cdot)$ is within $[-\pi/2, \pi/2]$ for all $-T \le s \le T$. This is the case if

$$N \ge \rho^{-2} \approx (\gamma\sigma)^{-2}.$$

In other words, to approximate a Gaussian having a small variance σ, N must at least be as large as $N_0 \approx (\gamma\sigma)^{-2}$. The bound is quite tight for large σ, but is loose when σ is small. We empirically determined N_0 for certain values of σ for the case $T = 255$ (cf. Table 1). It turned out to be much lower than the estimate $(\gamma\sigma)^{-2}$ when σ is small. For a fixed setting of T (e.g., for grayscale images), this suggests the use of a lookup table for determining N_0 for small σ on-the-fly.

The above analysis leads us to an $O(1)$ algorithm for approximating the Gaussian bilateral filtering, where both the spatial and range filters are Gaussians. The steps are summarized in Algorithm 5.

To summarize, our approach are the following advantages:

• **Accuracy**. Our method is exact. It does not require the quantization of the range kernel, as is the case in [110, 108]. Moreover, note that the auxiliary images in our case have the same dynamic range as the input image irrespective of the degree N. This is unlike the situation in [107], where the dynamic range of the auxiliary images grow exponentially with the N. This makes the computations susceptible to numerical errors for large N.

Algorithm 5 Fast $O(1)$ bilateral filtering using Gaussian kernels

---

Input: Image $f(x)$ with dynamic range $[-T, T]$, σ_s^2 and σ_r^2 for the spatial and range filters.

1. Set $\gamma = \pi/2T$, and $\rho = \gamma \sigma_r$.

2. If $\sigma_r > \gamma^{-2}$, pick any large N. Else, set $N = (\gamma \sigma_r)^{-2}$, or use a look-up table to fix N.

3. Set up the images $h_n(x) = \exp\left(j\gamma(2n - N)f(x)/\rho\sqrt{N}\right)$ and $g_n(x) = f(x)h_n(x)$, and the coefficients $d_n(x) = 2^{-N}\binom{N}{n}\exp\left(-j\gamma(2n-N)f(x)/\rho\sqrt{N}\right)$.

4. Filter every $h_n(x)$ and $g_n(x)$ with a Gaussian of variance σ_s^2 to get $\overline{h}_n(x)$ and $\overline{g}_n(x)$.

5. Set $\tilde{f}(x)$ as the ratio of $\sum_{n=0}^{N} d_n(x)\overline{g}_n(x)$ and $\sum_{n=0}^{N} d_n(x)\overline{h}_n(x)$.

Return: Bilateral filtered image $\tilde{f}(x)$.

• **Speed**. Besides having $O(1)$ complexity, our algorithm can also be implemented in parallel. This allows us to further accelerate its speed.

• **Approximation property**. Trigonometric functions yield better (local) approximation of Gaussians than polynomials. In particular, we showed that by using a particular class of raised cosines, we can obtain much better approximations of the Gaussian range kernel than that offered by the Taylor polynomials in [107]. The final output is artifact-free and resembles the true output very closely. The only flip side of our approach (this is also the case with [107], as noted in [108]) is that a large number of terms are required to approximate very narrow Gaussians over large intervals.

Implementation. We implemented the proposed algorithm for Gaussian bilateral filtering in Java on a Mac Pro 2.8 GHz dual core system as a ImageJ plugin. We used multi-threading for computing the spatial averages of the auxiliary images in parallel. A recursive $O(1)$ algorithm was used for implementing the Gaussian filter in space domain. The average time for processing a 512×512 grayscale image for a Gaussian range kernel of $\sigma_r = 80$ (independent of σ_s of the spatial Gaussian) was 440 and 310 milliseconds for the raised cosine and the polynomial approximation, respectively. The approximations are the ones shown in Figure 9.4. The execution of the latter is slightly lower than that of our method since it requires half the number of auxiliary images for a given degree. We note that the direct implementation of the filter (using discretized Gaussians supported on $[-3\sigma, 3\sigma]$) requires considerable time depending on the size of the spatial filter. For a 512×512 image, it is 4 seconds for σ_s as low as 3, and almost 14 seconds when $\sigma_s = 6$.

In Figure 9.5, we compare the processed outputs of the two algorithms with the direct implementation. Note that our result In Figure 9.5, we compare the processed outputs of the two algorithms with the direct implementation on the natural image of *Isha*. Note that our result resembles the exact output very closely. The result
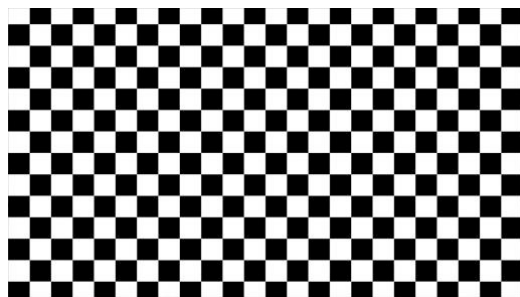
Figure 9.6: Test image of a checker pattern which is used for comparing the bilateral filters. The gray value is constant within each square (either 0 or 255). Artificial edges (transitions) were created between a given square and its four adjoining neighbors. This is used to test how well a given filter smooths the interiors of the squares, while preserving the edges at the same time.

obtained using the polynomial kernel, on the other hand, shows strange artifacts (see inset zoom). The difference is also clear from the standard deviation of the error between the exact output and the approximations.

9.2.3 Denoising experiments

To demonstrate the working of the space-variant bilateral filter, we apply it for image denoising. The main steps of our denoising algorithm are as follows.

- We compute the local energy $\mathcal{E}(x)$ as the weighted norm of the gradient $\mathbf{g}(x)$. We use the formula

$$\mathcal{E}(x) = \left[\int w(y)(\mathbf{g}^T\mathbf{g})(x-y)\,dy\right]^{1/2}.$$

This, in turn, is used to determine the scale $a(k)$ of the box spline at location k. We first fix the minimum and maximum scale to be a_{min} and a_{max} (these are set proportional to the noise variance). The rule for setting the scale is then given by

$$a(k) = (1-\lambda)a_{min} + \lambda a_{max}$$

where $\lambda = 1/(1+\mathcal{E})$. This ensures that the scales are always with the range $[a_{min}, a_{max}]$. This rule pushes the scale towards a_{min} when $\mathcal{E}$ gets large, and towards a_{max} for when $\mathcal{E}$ is small. In effect, we apply large spatial filters (more smoothing) in the homogenous regions, where we have less structural energy. On the other hand, we apply smaller spatial filters (less smoothing) close to edges, where the energy is typically higher.
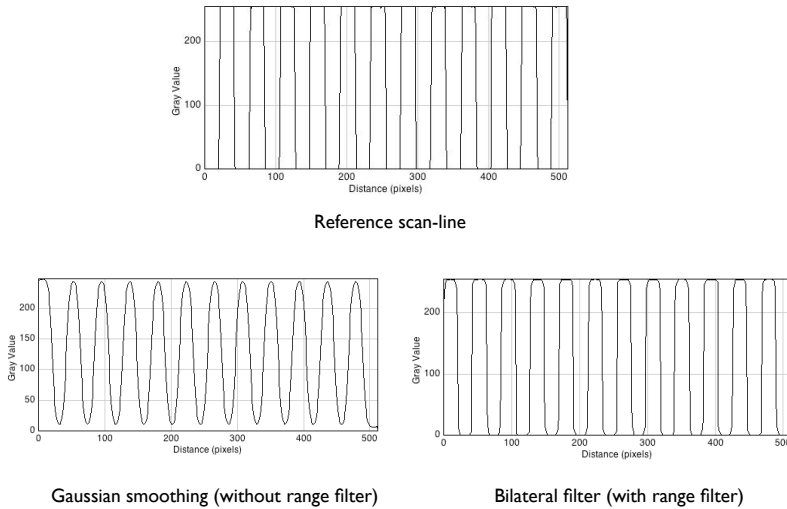
Reference scan-line

Gaussian smoothing (without range filter)   Bilateral filter (with range filter)

Figure 9.7: Comparison of the horizontal scan profiles of the smoothed images obtained by applying different filters to the image shown in Figure 9.6. The proposed space-variant (adaptive) bilateral filter is seen to preserve edges better than the standard bilateral filter.

- The auxiliary images $\varphi_i(x)$ and $\psi_i(x)$ are pre-integrated using running-sums.

- The parameters of the finite-difference mesh are computation using $a(k)$. The same finite-difference mesh is applied to the pre-integrated images to obtain the space-variant averages of the auxiliary images. We have the crucial advantage that the samples of the interpolating ZP function (required for the finite-difference, cf. Chapter 8) needs to be computed only once per pixel.

- The space-variant averages of the auxiliary images are combined using (9.7) to obtain the filtered image.

We now apply the space-variant bilateral filter for denoising synthetic and real images corrupted with additive Gaussian noise. We compare the results with those obtained using isotropic Gaussian filter and the standard bilateral filter. The algorithm was implemented in Java on a Macintosh 2.8 GHz Intel dual-core system. The execution time of our denoising algorithm is roughly 500 milliseconds for a 512×512 image. This included the time for computing the energy map and the scale vectors, the five running-sums, the interpolated samples, and the finite-differences. We use the raised-cosine range kernel for all the experiments.

We begin with the test image of *Checker Patterns* shown in Figure 9.6. To access the superiority of proposed space-variant filter, we apply it to this sythetic image. As reference, we also apply the isotropic Gaussian filter and the standard bilateral

Corrupted Image, PSNR=14.33 dB

(a) Gaussian smoothing, PSNR=14.98 dB

(b) Gaussian bilateral filter, PSNR=18.29 dB

(c) Adaptive bilateral filter, PSNR=19.47 dB
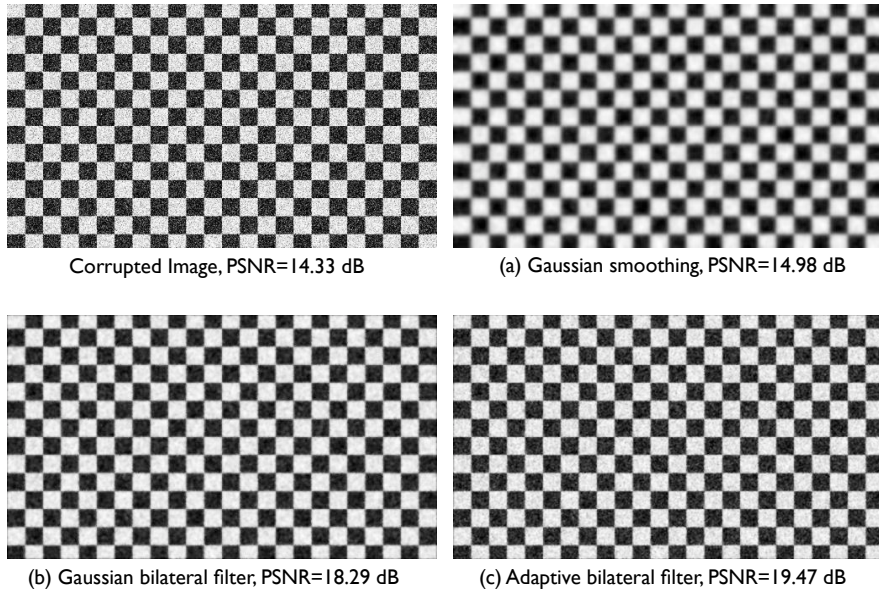
Figure 9.8: Cancellation of additive Gaussian noise. The test image in Figure 9.6 has been corrupted with white Gaussian noise. The figure shows the outputs obtained by applying various filters to this corrupted image. Notice the difference in the performance of the filters along the edges of the squares. The shrinkage derived from the range kernel of the bilateral filter is successful in reducing averaging across edges. The improvement is also evident from the respective PSNRs. The parameters of the filters have been tuned to optimize the PSNR.

filter to the same image. We use Gaussians of the same size for the isotropic and the bilateral filter. For the space-variant filter, we use this to set the size of the largest Gaussian a_{max}. We set $a_{min} = a_{max}/2$. The scan profiles of the denoised images are shown in Figure 9.7. It is clearly evident from the plots that the standard bilateral filter preserves the edges significantly better than Gaussian smoothing. The space-variant filter also yields a higher PSNR.

We now perform the following denoising experiment on the test image. We corrupt the image by adding white Gaussian noise of a specified variance. This is shown in Figure 9.8. It also shows the outputs of the three denoisng schemes. We used Gaussians of the same size for isotropic smoothing and the standard bilateral filter. The size of Gaussians were tuned to optimize the PSNR of the three filters. While the noise content of the space-variant filter is marginally more than that of the other two filters, it tends to provide better edge preservation than the standard bilateral filter, and of course, significantly better than the isotropic Gaussian filter. The improvement is also clear from the output PSNRs.

Finally, we compare the results obtained on the fluorescence image shown in Figure 9.2. We corrupt the image using white Gaussian noise. The denoising results are shown in Figure 9.9. In this case, the size of the Gaussians for the three filters were manually adjusted to maximize the PSNR of the output. As before, the improvement of the PSNR obtained using our space-variant filter is higher. Our filter provides better preservation of structures than the standard bilateral filter, at the expense of some additional background noise (in comparison to the isotropic Gaussian filter).

Appendix A

By Taylor's theorem (with remainder), we have $\cos(x) = 1 - x^2/2 + x^4 \cos(\xi)/24$, where ξ is a number between 0 and x. That is, $\cos(x) = 1 - x^2/2 + O(x^4)$. After some computation, it can be show that, for some constant c (independent of s and N),

$$\left[\cos\left(\frac{\gamma s}{\sqrt{N}}\right)\right]^N = \left(1 - \frac{\gamma^2 s^2}{2N}\right)^N + O(N^{-1}s^4) \qquad \left(|x| < c\sqrt{N}\right).$$

Using the well-known fact that $(1 + x/m)^m$ approaches $\exp(x)$ as m gets large, we get (9.8).

Corrupted Image, PSNR=14.93 dB

(a) Gaussian smoothing, PSNR=17.54 dB

(b) Gaussian bilateral filter, PSNR=17.86 dB

(c) Adaptive Gaussian bilateral filter, PSNR=18.25 dB
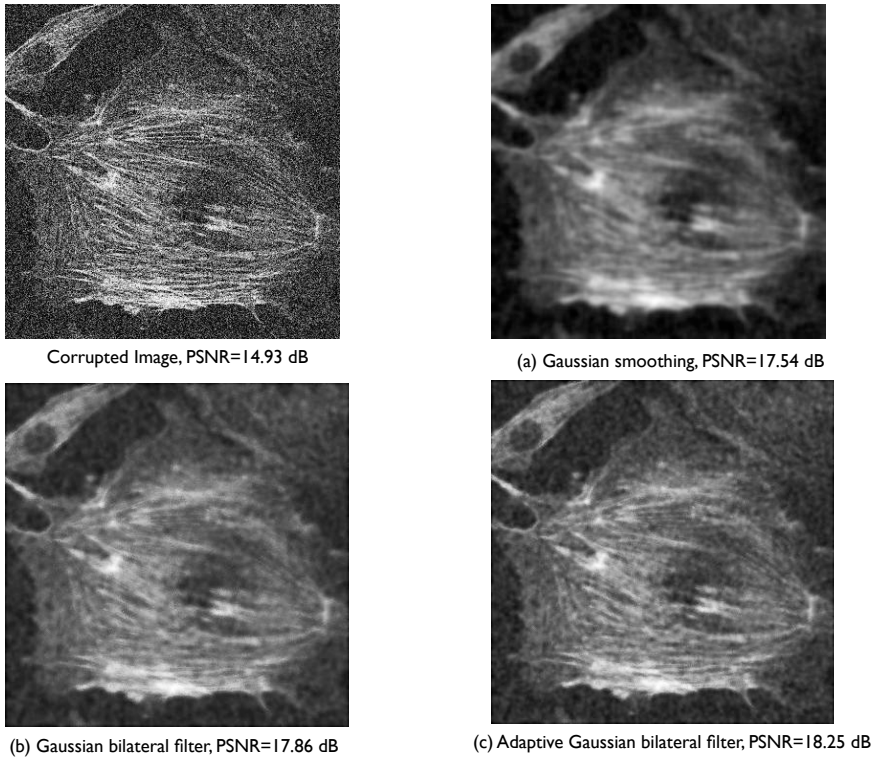
Figure 9.9: Results on real data. We use the biological image shown in Figure 9.2 and corrupt it with additive Gaussian noise. The denoising results shown in the figure have been obtained by manually optimizing the PSNRs of the three filters. Notice that the standard bilateral filter and our space-variant filter provide roughly the same amount of noise attenuation (e.g., see background regions). The space-variant version of the filter, however, provides better preservation of the filament structures, besides improving its PSNR.

Chapter 10

Fast detection of cells using a continuously scalable detector

Abstract — In this chapter, we propose a fast algorithm for the detection of cells in fluorescence images. The algorithm provides a robust estimate of the number of cells, and their respective centers and radii. It relies on the rapid computation of intensity-based correlations between the cells and a near-isotropic detector.

The attractive features of our algorithm are its speed and accuracy. We can compute the correlation between a cell and detectors of various sizes using a fixed number of operations (independent of the size of the detectors). On the other hand, we can continuously control the center and the radius of the detector to derive a precise estimate of the position and size of the cell.

We provide experimental results on both simulated and real data to demonstrate the speed and accuracy of the algorithm[1].

10.1 Intoduction

THE segmentation of cells is often an important pre-processing step for the quantitative analysis of microscopic images. For example, routine image analysis tasks in bio-microscopy, such as counting of cells, study of their spatial organization and the distribution of fluorescence signals on the nuclei, require a precise delineation of the cell boundaries.

In fluorescence imaging, the cells (or the cell nuclei) appear as bright blobs on a

---

[1]This chapter is based on the article [111]: K. N. Chaudhury, Zs. Püspöki, A. Muñoz-Barrutia, D. Sage, M. Unser, "Fast detection of cells using a continuously scalable Mexican-hat-like template," Proc. Seventh IEEE International Symposium on Biomedical Imaging, pp. 1277-1280, 2010.

dark background. The difficulty, however, is that only a limited amount of laser excitation can be used in order to avoid excess photobleaching. As a result, the images are often corrupted with large amounts of shot noise [112]. Added to this, there is typically the problem of uneven illumination, where the intensities within the cell are significantly varying. This makes it difficult to segment the cells using a single global threshold. Adaptive thresholding methods, often combined with region growing, have commonly been used to circumvent this problem [113, 114]. More recently, accurate and sophisticated segmentation algorithms based on level-sets and graph-cuts have been proposed for this task [115, 116]. The present limitations of such methods are the difficulty to automate them, and their slow-to-moderate computational speed. For certain applications, especially those related to high-throughput screening, it is absolutely necessary to adopt techniques that are fast and fully automated. Simple but efficient detection methods have been proposed to suit these requirements [117, 118].

In this Chapter, we propose a simple and fast algorithm for the detection of round cells in fluorescence images based on linear filtering. At the core of our algorithm is a fast technique for computing correlations between cells and a scalable detector, which is based on the fast algorithm for space-variant filtering proposed in Chapter 8. In the present context, we are interested in the "isotropic" variant of the radially-uniform box splines which were shown to resemble the isotropic Gaussians. A popular means of approximating the Laplacian-of-Gaussian (LoG) is using the difference of Gaussians. In our case, we realize a LoG-like detector using the difference of two near-isotropic box splines. These are then used as templates for detecting round cells in fluorescence images, and for estimating their positions and radii.

10.2   Scalable Laplacian-of-Gaussian detector

We briefly recall the construction of the isotropic variants of the radially-uniform box splines. The detector construction is considered next.

10.2.1   Approximation of isotropic Gaussian

The particular box splines that we are interested in are realized through the convolution of four "uniformly-rotated" box functions of identical width. Using the notations of Chapter 8, we can write this box spline as

$$\beta_r(\boldsymbol{x}) = (\varphi_{r,0} * \varphi_{r,\pi/4} * \varphi_{r,\pi/2} * \varphi_{r,3\pi/4})(\boldsymbol{x}) \tag{10.1}$$

where r is the width of the box functions.

This, in fact, can be seen as an improvement over the standard separable construction

$$\phi_r(\boldsymbol{x}) = (\varphi_{r,0} * \varphi_{r,0} * \varphi_{r,\pi/2} * \varphi_{r,\pi/2})(\boldsymbol{x}), \tag{10.2}$$
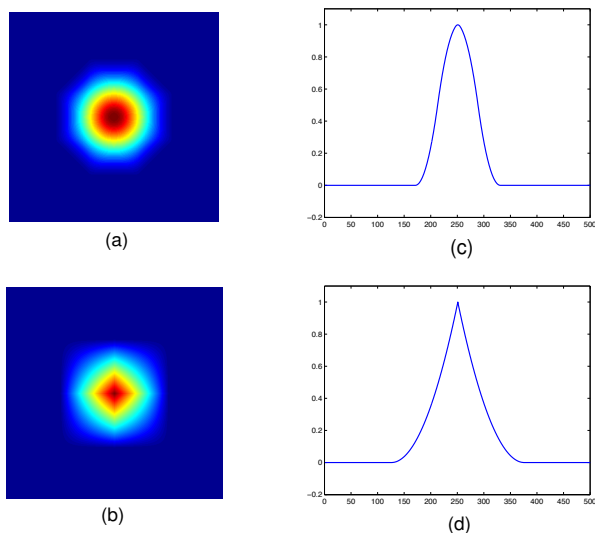
Figure 10.1: Intensity distributions of (a) $\beta_r(\boldsymbol{x})$ and (b) $\phi_r(\boldsymbol{x})$; (c) and (d): Respective scan profiles along $\pi/8$.

which uses box functions solely along the horizontal and the vertical directions. Although $\beta_r(\boldsymbol{x})$ and $\phi_r(\boldsymbol{x})$ have the same order (a total of four box functions in either case) the "rounding-effect" of the box functions placed along the diagonals tends to make the former more isotropic. Indeed, judging by the shape of the support and the distribution of the intensity of (10.1) and (10.2) shown in Figure 10.1, the former clearly looks more Gaussian-like than the latter. A quantitative justification of this fact is obtained through the computation of the *isotropy index*

$$\varrho = \frac{1}{2\pi\|f\|^2} \int_0^{2\pi} \langle R_\theta f, f \rangle \, d\theta$$

where R_θ is the operator that rotates a function by angle θ. This index measures the rotational symmetry of a non-negative function by correlating it with its rotated versions. Clearly, $\varrho = 1$ if the function is truly isotropic. In general, the higher the value of ϱ, the better is the isotropy of the function. The index for (10.2) was found to be 98.8%, while a higher index of 99.7% was recorded for (10.1).

10.2.2 Detector specification and characteristics

The Laplacian-of-a-Gaussian (LoG), also known as the Mexican-hat, is widely used for detecting radial singularities in images. In practice, the LoG is often approximated by the difference of Gaussians, which is easier to implement. In our case, we
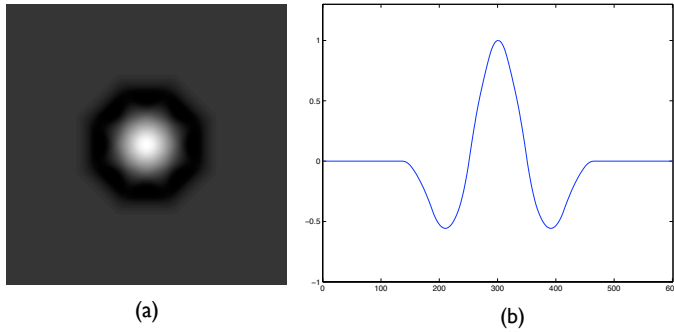
(a)   (b)

Figure 10.2: (a) Intensity distribution of the LoG-like detector $\psi_r(x)$; (b) Scan profile along $\pi/8$.

approximate the LoG by the difference of a narrow and a wide box spline,

$$\psi_r(x) = \beta_r(x) - \beta_{\sqrt{2}r}(x), \tag{10.3}$$

where $r > 0$ is a real-valued parameter that controls the scale (*radius*) of the detector. Figure 10.2 shows the intensity distribution of $\psi_r(x)$ and its profile along a scan-line passing through the origin.

Note that, by construction, the total mass of $\beta_r(x)$ does not depend on r. Therefore,

$$\int \psi_r(x)\, dx = 0 \qquad (r > 0).$$

This means that the detector tends to suppress uniform-intensity regions, while it gives a large response for singularities along the radial direction. To make this more precise, we consider the ideal blob function with centre x_c and radius R:

$$b_R(x; x_c) = \begin{cases} 1 & \text{for } 0 \leq \|x - x_c\| \leq R \\ 0 & \text{otherwise.} \end{cases} \tag{10.4}$$

The radial singularity in this case is along the circumference of the blob. Assume that (10.3) is centered at x_c. The corresponding detector response is given by

$$M_r(x_c) = \int b_R(x; x_c)\psi_r(x - x_c)\, dx.$$

We now vary its radius r. It turns out that $M_r(x_c)$ is a smooth unimodal function of r. Its peak is located at γR, where $\gamma \approx 1.33$ is a calibration constant. This unimodal response is exactly due to the cancellation that takes place between the constituent box splines when $r \leq \gamma R$. The cancellation keeps increasing as r goes from γR to

zero resulting in the progressive drop in the response. On the other hand, when $r > \gamma R$, no cancellation occurs. The decay of $M_r(x_c)$ in this regime is due to the $1/r^4$ normalization of the constituent box splines.

Although (10.4) is a rather idealized model for cells (or nuclei) in real images, the responses obtained turn out to be very similar for real fluorescence images. For instance, Figure 10.3 shows the response obtained for one of the cells in the fluorescence image in Figure 10.5.

10.2.3 Computational aspects

Before presenting the algorithm, we elaborate on the computation of the quantity

$$M_r(x_c) = \int f(x)\psi_r(x - x_c)\,dx \tag{10.5}$$

where $f(x)$ is discrete,

$$f(x) = \sum_{n \in \mathbb{Z}^2} f[n]\delta(x - n).$$

A fast and efficient evaluation of (10.5) will turn out to be the workhorse of our cell detection algorithm, which requires the rapid evaluation of $M_r(x)$ at several image positions, and at different radii per position.

From (10.3), it is clear that $M_r(x)$ can be expressed as the difference of two averages,

$$M_r(x) = A_r(x) - A_{\sqrt{2}r}(x) \tag{10.6}$$

where

$$A_r(x) = \int f(y)\beta_r(y - x)\,dy. \tag{10.7}$$

The crucial point is that we can exactly determine (10.7) using the algorithm of Chapter 8. We briefly describe the steps of the algorithm in the present setting.

(1) **Pre-integration**. We pre-integrate the discrete image $f[n] = f[n_1, n_2]$ along the four cardinal directions using running-sums. These are efficiently implemented using recursion.

(i) Horizontal, $g_0[n_1, n_2] = \sum_{k=0}^{\infty} f[n_1 - k, n_2]$.

(ii) First-diagonal, $g_{\pi/4}[n_1, n_2] = \sqrt{2}\sum_{k=0}^{\infty} g_0[n_1 - k, n_2 - k]$.

(iii) Vertical, $g_{\pi/2}[n_1, n_2] = \sum_{k=0}^{\infty} g_{\pi/4}[n_1, n_2 - k]$.

(iv) Second-diagonal, $g[n_1, n_2] = \sqrt{2}\sum_{k=0}^{\infty} g_{\pi/2}[n_1 + k, n_2 - k]$.

(2) **Finite-difference**. For a given position x and radius r, we compute $A_r(x)$ by taking a finite-difference of an interpolation of the pre-integrated image. In particular, we use the formula

$$A_r(x) = \sum_{i=0}^{15} w_i\, G(x + \tau - x_i)$$

where

- The tuple (q_4, q_3, q_2, q_1) is the binary representation of the indices $0 \leq i \leq 15$, so that it takes values from $(0,0,0,0)$ to $(1,1,1,1)$.

- $w_i = (-1)^{q_1} \cdots (-1)^{q_4}(1/r^4)$ and $x_i = r(q_1 u_0 + q_2 u_{\pi/2} + q_3 u_{3\pi/4} + q_4 u_{\pi/4})$.

- $\tau = (\tau_1, \tau_2)$ where $\tau_1 = (r-1)/2$ and $\tau_2 = (\sqrt{2}r + r - 3)/2$.

- $G(x)$ is the interpolation of $g[n]$ using the box spline $\beta_{\mathrm{ZP}}(x)$,

$$G(x) = \sum_{n \in \mathbf{Z}^2} g[n]\beta_{\mathrm{ZP}}(x - n).$$

We recall that the samples $G(x + \tau - x_i)$ can be rapidly evaluated by taking advantage of the finite support and the piecewise-quadratic structure of $\beta_{\mathrm{ZP}}(x)$; see Appendix F of Chapter 8.

Once the pre-integration is over, $M_r(x)$ is evaluated simply by computing (10.7) for radii r and $\sqrt{2}r$ using step 2, and then applying (10.6). The crucial aspect of the above computation is that the number of operations (modulo the mild interpolations) required in step 2 does not depend on r. Hence, the cost of computing $M_r(x)$ does not vary with the size of the detector. This $O(1)$ complexity is clearly a significant improvement over a naive implementation of (10.5) involving the discretization of the detector and the integral, which would require $O(r^2)$ computations.

An alternative would be to use some fast filtering technique to perform the correlations with a fixed template over the entire image. The advantage of the present computational strategy is that we can concentrate exclusively on refining the matching around some location, which speeds up the whole process considerably.

10.3   Fast detection of cells

10.3.1   The image model

We model the image $f(x)$ as the superposition of circular blobs $b_{R_i}(x; x_i)$ of unknown radius R_i and intensity α_i, along with background noise $n(x)$:

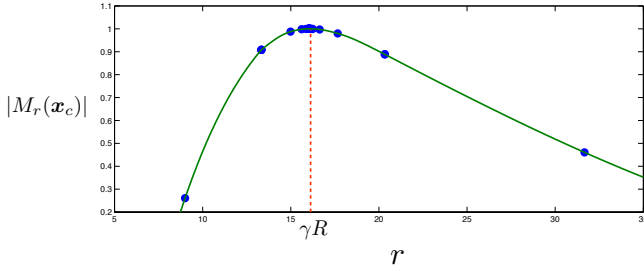$$f(x) = \sum_{i=1}^{N} \alpha_i\, b_{R_i}(x; x_i) + n(x). \tag{10.8}$$

Figure 10.3: A typical detector characteristic for one of the cells in Figure
10.5. The solid curve shows the variation of the detector response $M_r(\boldsymbol{x}_c)$
with radius r, where $\boldsymbol{x}_c$ is the exact cell center. The dots on this curve
correspond to the successive estimates of the optimal point $\gamma R^\star$ obtained
from the golden section search.

We consider a more realistic model of the blobs $b_{R_i}(\boldsymbol{x}; \boldsymbol{x}_i)$ than the one in (10.4).
We do assume each cell to be localized within a disk of radius R_i, but we do not
assume the blobs to have the same height or uniform intensity distributions.

The proposed algorithm estimates (without a priori knowledge) the number of cells
(N), and the centre ($\boldsymbol{x}_i$) and radius (R_i) of each cell. The only assumption used is
that the radii of the cells are bounded, that is, $R_{\min} \leq R_i \leq R_{\max}$, where $R_{\min}$ and
$R_{\max}$ are provided as user inputs. Our approach involves the joint-estimation of the
centers and the radii, whereby we first sample the values of $M_r(\boldsymbol{x})$ at discrete image
positions and radii to obtain a coarse-to-fine estimate of the center $\boldsymbol{x}_i$. Once this is
determined, we proceed to estimate the radius by optimizing $M_r(\boldsymbol{x}_i)$.

10.3.2   The algorithm

The main steps of the algorithm are as follows.

(1) **Coarse estimation of the centers, rejection of background points**. To obtain
a coarse estimate of the centers we restrict the potential cell centers, aka the *candidate
points*, to a coarse lattice of resolution $\lceil 1.8R_{\min} \rceil \times \lceil 1.8R_{\min} \rceil$. This particular choice
of resolution ensures that at least one lattice point intersects every cell. Let us
denote these candidate points by $\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_P$. We place them in a list $\mathscr{L}$. In general,
$P \gg N$, but is small compared to the size of the image.

At every $\hat{\boldsymbol{x}}_i$, we compute $M_r(\hat{\boldsymbol{x}}_i)$ at $r_k = R_{\min} + k(R_{\max} - R_{\min})/4$ for $k = 1, 2$, and
3. We set

$$M_i = \max_k |M_{r_k}(\hat{\boldsymbol{x}}_i)|.$$

We remove those $\hat{\boldsymbol{x}}_i$ (typically the background pixels) from $\mathscr{L}$ for which M_i is
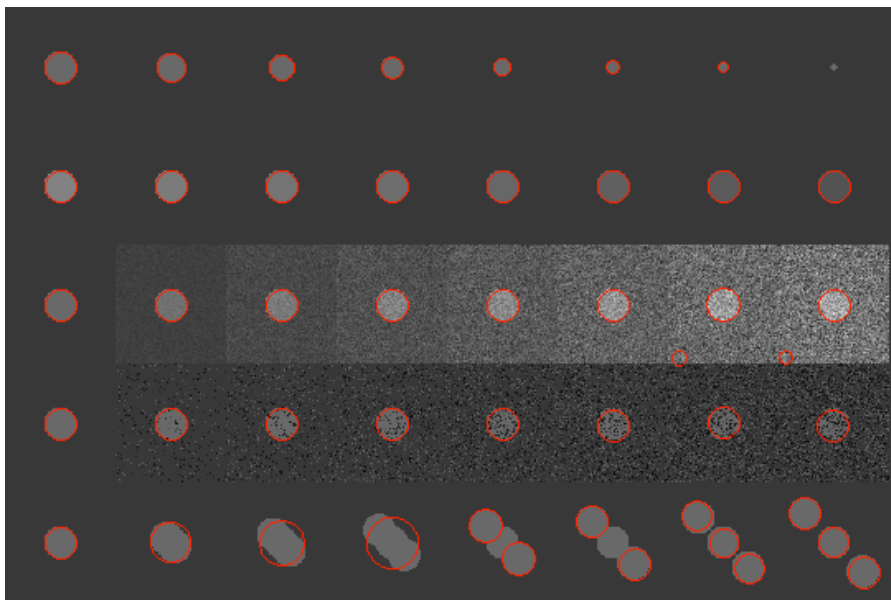
Figure 10.4: Detection result on simulated data. **Row** 1: Blobs of different size; **Row** 2: Blobs of different intensities; **Row** 3: Identical blobs corrupted with different levels of additive Gaussian noise; **Row** 4: Identical blobs corrupted with different levels of speckle noise; and **Row** 5: Cluster of identical blobs with varying margin of separation.

smaller than a specific threshold ε. We sort the remaining points into $\hat{x}_1, \ldots, \hat{x}_K$ using the criterion that $\hat{x}_j$ comes before $\hat{x}_k$ if and only if $M_j \geq M_k$. Note that $K << P$. The reasoning here is that those $\hat{x}_i$ which are close to the actual centers x_i tend to generate larger responses than those that are further off. The above ordering places such points in the foremost part of $\mathcal{L}$.

(2) **Fine estimation of the center and the radius**. We set $N = 0$ and $i = 1$. We visit the foremost candidate point in $\mathcal{L}$, place an appropriate window W around it on the image lattice. Similar to the previous phase, we compute the maximum response

$$M(\xi) = \max_k |M_{r_k}(\xi)|$$

at every $\xi \in W$. We use the result of this search over a finer lattice to select the current center as the point corresponding to the local maxima, that is,

$$x_i = \arg\max_{\xi \in W} M(\xi).$$

Having estimated the current centre, we use the unimodal characteristics of the detector (cf. Figure 10.3), and the hypothesis that $R_{\min} \leq R_i \leq R_{\max}$, to estimate the

radius. In particular, we set

$$R_i = \gamma^{-1} \arg\max \left\{ |M_r(\boldsymbol{x}_i)| : \gamma R_{\min} < r < \gamma R_{\max} \right\}.$$

We perform the above optimization using the golden-section search [119]. This is an efficient algorithm for finding the extremum of a unimodal function, where one localizes the extremum by successively shrinking the size of the interval within which this is known to exist.

We now have the i-th cell with centre $\boldsymbol{x}_i$ and radius R_i. We increment N to $N+1$. We then proceed to remove the points from $\mathscr{L}$ which belong to the region of the detected cel, i.e., the region $\{\boldsymbol{x} : \|\boldsymbol{x} - \boldsymbol{x}_i\| \leq R_i\}$. We increment i to $i+1$, and repeat the above process, namely, the estimation of the centre and radius of the $(i+1)$-th cell, and the crucial speed-up step involving the truncation of $\mathscr{L}$.

(3) **Convergence.** The iteration is terminated if either the list of candidate points is exhausted, or if the maximum projection goes below ε. The latter typically occurs when all the bright cells have been detected, leaving behind the cells of weak intensity and the background pixels.

The pseudocode for the algorithm is given in Algorithm 6. We have used the following notations. For a given position $\boldsymbol{x}$, we denote the maximum response over the range $[R_{\min}, R_{\max}]$, evaluated at fixed radii $r_1, \ldots, r_n$, by

$$\text{MaxResponse}(\boldsymbol{x}) = \max_{1 \leq i \leq n} M_{r_i}(\boldsymbol{x}).$$

Given an ordered list $\mathscr{L}$, we use Header($\mathscr{L}$) to denote the first element of $\mathscr{L}$.

10.4   Experiments

To evaluate the performance of the proposed algorithm, we simulated a single 800×400 image using different variations of the cell model in (10.8). This is shown in Figure 10.4. The idea was to study the effect of the following on the performance of the algorithm:

- The size and intensity of the blobs (rows 1 and 2),

- The presence of ambient noise (rows 3 and 4),

- The separation between the cells (row 5).

The respective SNRs of the images in row 3 (Gaussian noise) and row 4 (speckle noise) were

$$+\infty, \ 37.5, \ 31.5, \ 27.8, \ 25.4, \ 23.4, \ 21.8, \ 20.6 \text{ dB},$$

Algorithm 6 Cell detection algorithm.

1. **Input**: Image $f(x)$; bounds R_{min}, R_{max}; tolerance ε; window W; γ.

2. **Initialization** of $\mathscr{L}$, the list of potential centers.
   (i) Fix lattice $\mathscr{L} = [x_1, \ldots, x_P]$ of appropriate resolution.
   (ii) Determine $\text{MaxResponse}(x_1), \ldots, \text{MaxResponse}(x_P)$.
   (iii) If $\text{MaxResponse}(x) < \varepsilon$, then remove x from $\mathscr{L}$.
   (iv) Sort remaining points in $\mathscr{L}$ using the order

   $$x > y \text{ if and only if } \text{MaxResponse}(x) \geq \text{MaxResponse}(y) \qquad (x, y \in \mathscr{L}).$$

   (v) Set $N = 0$; initialize list of blobs, $\text{BLOBS} = []$.

3. **do** until convergence
   (i) $x \longleftarrow \text{Header}(\mathscr{L})$.
   (ii) Center W at x, and set $\hat{x} \longleftarrow \arg\max \left\{ \text{MaxResponse}(\xi) : \xi \in W \right\}$.
   (iii) Compute $\hat{R} = \gamma^{-1} \arg\max \left\{ M_r(x) : \gamma R_{min} \leq r \leq \gamma R_{max} \right\}$.
   (iv) Update: $\text{BLOBS} \longleftarrow [\text{BLOBS}, (\hat{x}, \hat{R})]$; $N \longleftarrow N + 1$.
   (v) Remove points from $\mathscr{L}$ which are in the disk with center $\hat{x}$ and radius R.
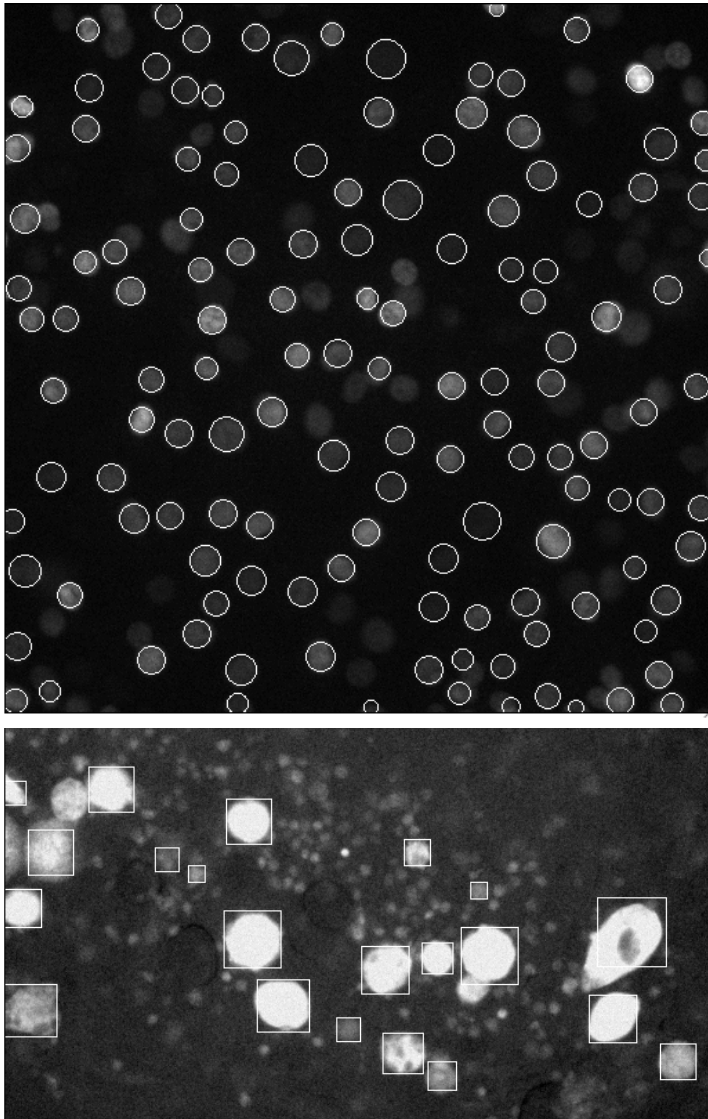4. **Output**: Number of blobs N, and the list BLOBS.

Figure 10.5: Top: Snapshot of moving cell nuclei (shown using white circles) detected from an image sequence (Courtesy of C. Dibner, University of Geneva). **Bottom**: Bright stem cells (shown using white bounding boxes with a cross) detected from a fluorescence image using our algorithm (Courtesy of N. Garin, ISREC, EPFL).

and
$$+\infty,\ 40.1,\ 37.2,\ 35.8,\ 34.6,\ 33.7,\ 33.3,\ 32.5\ dB.$$

The detected blobs are shown in Figure 10.4 using red circles. A near-exact estimate of the positions and radii was obtained for the blobs in the first four rows. The error in localization was within 1 pixel, and that for the radius was within 1%. There were however two false detections in presence of noise, and the algorithm failed to detect the smallest blob in row 1and the overlapping blobs in row 5. The contiguous blobs in this row 5 were detected as a single object.

Figure 10.5 shows the results of our detection algorithm applied to real fluorescence images. The top Figure shows a NIH3T3 cell line (mammalian cells for circadian cycle analysis) of size 512×512, which stably expressed the nuclear fluorescent protein under circadian *Reverba* promoter regulation. The second fluorescence image (cropped to 512×312 pixels) was obtained from an experiment on the migration and proliferation of stem cells. We used $R_{min} = 3, R_{max} = 32$ and $W = 3R_{min}$ in either case. Most of the bright blobs were detected in both the images, including the slightly elongated cells in the second image. The very faint nuclei in the first image and the tiny blobs in the second image were not detected. The average execution time for the NIH3T3 and the stem cell image was 700 and 400 milliseconds, respectively. Note that the detection results for the stem cells shows that the algorithm does reasonably well even when the cells are not particularly round.

The algorithm was implemented in Java on a Macintosh 2.8 GHz Intel dual-core system. We set $R_{min} = 3, R_{max} = 20$, and $W = 3R_{min}$. The execution time for the 512×512 image was about 600 milliseconds.

Chapter 11

Conclusion

11.1 Summary

IN this thesis, we addressed the construction of Gabor-like wavelets, and smoothing kernels based on the Gaussian. We constructed a family of spline wavelets which provided arbitrary close approximations of the Gabor function. On the other hand, we introduced a family of compactly supported box splines to approximate the optimally-localized Gaussian, both isotropic and anisotropic. The attractive feature of these spline wavelets and kernels was that we are able to develop fast and efficient algorithms to implement the associated transforms.

The Gabor function has the unique property of being optimally localized in both space and frequency. In Chapters 2-6, we showed how Gabor-like wavelets could be constructed within the framework of multiresolution analysis of Mallat and Meyer. The central theme of our investigation was the connection between a fundamental singular integral transform, the Hilbert transform, and wavelet bases derived by the dilation and translation of a single oscillating prototype having a prescribed number of vanishing moments. In particular, the construction of the Gabor-like complex wavelets was motivated by the observation that the real and the imaginary components of the Gabor function form an approximate Hilbert transform pair. On the other hand, it was known that a certain class of spline wavelets, the B-spline wavelets, converged to the real part of the Gabor function with the increase of their order. The goal then was to realize the Gabor-like wavelets by applying Hilbert transform to the B-spline wavelets and complexifying the resulting Hilbert transform pairs of wavelets. This led us to address the more general problem of constructing Hilbert transform pairs of wavelets within the framework of multiresolution analysis.

To begin with, we provided a self-contained exposition of the local and global properties of the Hilbert transform that are subsequently used in the thesis in Chapter 2. We also identified the complete class of integral transform, the fractional

Hilbert transforms, that are unitary (energy-preserving) and exhibit the fundamental invariances of the Hilbert transform with respect to translations and dilations of the signal. A particular directional extension of the fractional Hilbert transforms was also introduced.

Our first basic observation was that the Hilbert transform could be seamlessly integrated into the multiresolution framework of wavelets thanks to its invariance to dilations and translations, and its energy-preserving nature. It is in connection with the interaction of the transform with scaling functions (which are low-pass by construction) and compactly supported wavelets, that we faced certain technical difficulties. We observed that the Hilbert transform of a scaling function was no longer a valid scaling function, and the Hilbert transform of a compactly supported wavelet was necessarily of infinite support, with often a very slow decay; e.g., the Hilbert transform of the Haar wavelet had only a modest quadratic decay. We showed that the former problem can be dealt with through a careful design of the wavelet multiresolution, in which the Hilbert transform is applied only on the wavelet (which is oscillatory by nature) and never explicitly on the scaling function. As far as the latter pathology is concerned, we showed how this can be overcome simply by considering a classes of well-localized wavelets having sufficient number of vanishing moments. The Hilbert transform of such wavelets were shown to have comparable vanishing moments, and, importantly, a decay proportional to the number of vanishing moments of the original wavelet.

Guided by the above considerations, we proceeded to formulate a recipe for constructing Hilbert transform pairs of wavelet bases. The key components of this construction was the B-spline factorization theorem, and the observation that the Hilbert transform of a B-spline could be expressed as the "discrete-filtered" version of a larger class of so-called fractional B-splines having identical approximation order. The former allowed us to express a sufficiently well-behaved scaling functions in terms of B-splines, while the latter allowed us to transfer the slow decay of the transform onto a discrete filter which was eventually applied on the oscillating wavelet filter. Using this framework, we were able to identify pairs of (ordered) spline multiresolutions where the associated wavelets formed a Hilbert transform pair. It was shown that the complex wavelet obtained by combining the Hilbert pairs converged to a Gabor function with the increase in order. Motivated by the construction of Kingsbury et al. [26], we next extended the construction to higher dimensions using the directional Hilbert transform and tensor-products wavelets. This resulted in a system of complex wavelets that closely resemble the directional Gabor functions. We also proposed an efficient numerical implementation of the Gabor-like wavelet transforms using Mallat's filterbank algorithm.

We next investigated the reconstruction mechanism (the functional representation) associated with the Gabor-like transform in arbitrary dimensions. The functional representation was arrived at based on the role of played by the fractional Hilbert transforms in the reconstruction mechanism, their invariances to translations and dilations, and their modulation property (Bedrosian theorem). In particular, we highlighted the resemblance of the representation with the classical widowed-Fourier

representation. This formal connection allows us to understand the significance of the amplitude and phase information associated with the transform.

In Chapter 7, we developed a coarse-to-fine stereo-matching algorithm by applying dynamic programming on the sub-sampled Gabor-like wavelet pyramid instead of the raw pixel intensities. The crucial feature of our pyramid was that it provided near translation-invariance at the cost of moderate redundancy. The translation-invariance proved to be absolutely essential for encoding the local spatial translations between the stereo pair. Based on the specific Gabor-like form of our complex wavelets, we also provided a mathematical explanation of the near translation-invariance of our pyramid. From a computational standpoint, we showed that a significant reduction of the run time is achieved by our algorithm in comparison with the standard dynamic programming algorithm.

The focus of Chapters 8-10 was on different forms of space-variant filtering based on Gaussian-like kernels. In Chapter 8, we investigated the efficient realization of space-variant filters modeled on anisotropic Gaussians. The realization of the space-variant Gaussian filter is known to be computationally challenging, particularly when the size of filter is large. We demonstrated that it is possible to filter an image with a Gaussian-like box splines of varying size, elongation and orientation using a fixed number of computations per pixel (constant-time implementation). The associated algorithm was realized using simple pre-integrations and local finite-differences. We also developed a simple root-finding algorithm for controlling the anisotropy of the filter.

As an application of the Gaussian-like box splines and the associated filtering algorithm, we proposed two space-variant filtering strategies in Chapter 9. The first of these was inspired by anisotropic Gaussian diffusion. The space-variance in this case is in terms of the size, elongation, and orientation of the box splines. The other scheme was based on a space-variant form of the Gaussian bilateral filter. The spatial adaptability in this case was in terms of the size of the spatial Gaussian filter. We developed a constant-time algorithm for implementing this filter by approximating the variable spatial filter using isotropic box splines, and by approximating the fixed range filter using shiftable kernels. As an application in a different direction, we constructed box spline filters resembling the Laplacian-of-Gaussian in Chapter 10. Using this particular detector, and by appropriately modifying the algorithm in Chapter 8, we developed a fast "template-matching" algorithm for the detection of bright cells and nuclei in fluorescence images.

11.2   Future research extensions

We now discuss some possible directions along which the present research work can be extended.

- **Applications of the 2-dimensional Gabor-like wavelet transform**. We used the 1-dimensional Gabor wavelets for developing a coarse-to-fine stereo-matching

algorithm in Chapter 7. We have, however, not fully exploited the reversibility of the wavelet transform; that is, the possibility of reconstructing the original signal from the analysis coefficients. For example, we could apply the 2-dimensional transform for image processing applications such as denoising and deconvolution. This has already been investigated for certain class of wavelet transforms with promising results [120, 121]. We could make possible use of the shiftability of our Gabor-like wavelet transform for such applications. The other feature of 2-dimensional wavelet transform is that it exhibits better translation invariance along certain preferred directions. This could be used for directional analysis of image features such as those required in texture analysis and synthesis applications [122].

• **Gabor-like wavelet packets**. In wavelet packet decomposition, one analyzes the signal by passing it through more filters than the standard discrete wavelet transform [14]. The idea is to improve the frequency resolution of the standard wavelet transform by decomposing the detail coefficients along with the approximation coefficients. This was introduced by Coifman et al. by generalizing the link between multiresolutions and wavelets [123].

The distinctive difference of wavelet packets with the standard wavelet paradigm is that, as against the use of single mother wavelet, the latter uses different wavelets at different scales (wavelet dictionary). There is evidence to believe that, if we applied this scheme to the B-spline scaling function along with appropriate wavelet filters, the corresponding wavelets would asymptotically converge (for large orders) to Gaussian-modulated cosines having distinctive modulation frequencies. This suggests the possibility of constructing a secondary system consisting of the Hilbert transforms of the B-spline wavelets. The complexification of these two "matched" systems would result in a larger class of Gabor-like wavelets (Gabor-wavelet dictionary) having different modulation frequencies.

• **Extension of the space-variant filtering to 3-dimensions**. The possibility of extending the space-variant filter (introduced in Chapter 8) to volume data was not explored in the thesis. Extending the running-sums and the finite-differences to 3-dimensions is conceptually straightforward. In particular, one would have to integrate the image along the principal directions of the 3-dimensional lattice, and would have to use a larger mesh of size 2^N (N being the number of directions) for computing the finite-differences. The challenge, however, is the computational feasibility in that one would have to compute the 2^N taps of the mesh on-the-fly, and use high-precision arithmetic for manipulating the running-sums. The added challenge of designing a fast algorithm for controlling the anisotropy of the box splines using its scales also remains open.

• **Improvement of the stereo-matching algorithm**. This is still a lot of scope of improving the stereo-matching algorithm. We have discussed the various possibilities in detail in Chapter 7.

• **Applications of space-variant bilateral filters**. The fast bilateral filter proposed in Chapter 9 could be used for various problems in image processing, computer graphics, and computer vision. These includes denoising [97, 98, 99, 100], texture editing [101], demosaicking [102], optical-flow estimation [103, 104], and stereo matching [105, 106], to name a few. For example, we could use the space-variant bilateral filter for the "cost-aggregation" step of the stereo-matching algorithm. In general, it has been demonstrated that the standard bilateral filter gives superior performance for such applications.

• **Refinement of the cell-detection algorithm**. The cell-detection algorithm developed in Chapter 10 does reasonably better than the standard convolution-based template matching in terms of speed and accuracy of the detection. The drawback of the algorithm is that it is designed to handle only round cells using isotropic templates. We could instead use a larger class of elliptical templates obtained by taking the difference of two elliptical box splines of different size, but having same orientation and elongation. This would result in a more flexible set of elliptical detectors, which could be used to detect elliptical and round cells at the same time. The non-trivial part, however, is the design of the optimization algorithm where one would have to simultaneously optimize the detector response over three different degrees of freedom—the size, the elongation, and the orientation of the template.

Bibliography

[1] W. Heisenberg, "Uber den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik," *Zeitschrift für Physik*, vol. 43, pp. 172–198, 1930.

[2] D. Gabor, "Theory of communication," *J. Inst. Elect. Eng.*, vol. 93, pp. 429–457, 1946.

[3] J.G. Daugman, "Two-dimensional spectral analysis of cortical receptive field profile," *Vision Research*, vol. 20, pp. 847–856, 1980.

[4] M.J. Bastiaans, "Gabor's expansion of a signal into Gaussian elementary signals," *Proc. IEEE*, vol. 68, pp. 538–539, April 1980.

[5] O. Christensen, *Frames and Bases: An Introductory Course*, Birkhäuser Boston, 2008.

[6] C. de Boor, K. Höllig, and S. Riemenschneider, *Box Splines*, Springer-Verlag, 1993.

[7] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part I — theory," *IEEE Transactions Signal Processing*, vol. 41(2), pp. 821–833, 1993.

[8] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part II —Efficient design and applications," *IEEE Transactions Signal Processing*, vol. 41(2), pp. 834–848, 1993.

[9] L. Schumaker, *Spline Functions: Basic Theory*, Cambridge Mathematical Library, 2007.

[10] S. Mallat, "Multiresolution approximation and wavelet orthogonal bases of L^2," *Transactions of the American Mathematics Society*, vol. 315, pp. 69–87, 1989.

[11] Y. Meyer, *Wavelets and Operators*, Cambridge Studies in Advanced Mathematics, 1995.

[12] M. Unser, A. Aldroubi, and M. Eden, "A family of polynomial spline wavelet transforms," *Signal Processing*, vol. 30, no. 2, pp. 141–162, January 1993.

[13] M. Unser, A. Aldroubi, and M. Eden, "Polynomial spline signal approximations: Filter design and asymptotic equivalence with Shannon's sampling theorem," *IEEE Transactions on Information Theory*, vol. 38, pp. 95–103, 1992.

[14] S. Mallat, *A Wavelet Tour of Signal Processing*, San Diego, CA: Academic Press, 1998.

[15] P. Burt, "Fast filter transform for image processing," *Computer Graphics and Image Processing*, vol. 16(1), pp. 20–51, 1981.

[16] P. Burt and E. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31(4), pp. 532–540, 1983.

[17] J. Crowley, *A Representation for Visual Information*, Ph.D. thesis, Carnegie-Mellon University, Robotics Institute, Pennsylvania, 1981.

[18] A. Haar, "Zur theorie der orthogonalen funktionensysteme," *Mathematische Annalen*, vol. 69(3), pp. 331–371, 1910.

[19] M. Unser, A. Aldroubi, and M. Eden, "The L^2 polynomial spline pyramid," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15(4), pp. 364–379, 1993.

[20] M. Unser, A. Aldroubi, and M. Eden, "On the asymptotic convergence of B-spline wavelets to Gabor functions," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 864–872, March 1992.

[21] T. Blu and M. Unser, "A complete family of scaling functions: The (α, τ)-fractional splines," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. VI, pp. 421–424, April 6-10 2003.

[22] M. Unser and T. Blu, "Fractional splines and wavelets," *SIAM Review*, vol. 42, no. 1, pp. 43–67, March 2000.

[23] R.R. Coifman and D.L. Donoho, "Translation-invariant de-noising," *Lecture Notes in Wavelets and Statistics*, pp. 125–150, 1995.

[24] N.G. Kingsbury, "Shift invariant properties of the dual-tree complex wavelet transforms," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1221–1224, March 1999.

[25] I.W. Selesnick, "Hilbert transform pairs of wavelet bases," *IEEE Signal Process. Lett.*, vol. 8, no. 6, pp. 170–173, 2001.

[26] I.W. Selesnick, R.G. Baraniuk, and N.C. Kingsbury, "The dual-tree complex wavelet transform," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 123–151, November 2005.

[27] N.G. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," *Journal of Applied and Computational Harmonic Analysis*, vol. 10, no. 3, pp. 234–253, May 2001.

[28] P.S. Heckbert, "Filtering by repeated integration," *International Conference on Computer Graphics and Interactive Techniques*, vol. 20, no. 4, pp. 315–321, 1986.

[29] M. Wells, "Efficient synthesis of Gaussian filters by cascaded uniform filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 234–239, 1986.

[30] T.A. Poggio, V. Torre, and C. Koch, "Computational vision and regularization theory," *Nature*, vol. 317, pp. 314–319, 1985.

[31] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

[32] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *Proc. of the Sixth International Conference on Computer Vision*, pp. 839–846, 1998.

[33] K.N. Chaudhury and M. Unser, "On the shiftability of dual-tree complex wavelet transforms," *IEEE Transactions on Signal Processing*, vol. 58, pp. 221–232, 2010.

[34] E.M. Stein, *Singular Integrals and Differentiability Property of Functions*, Princeton University Press, 1970.

[35] L. Grafakos, *Classical and Modern Fourier Analysis*, Prentice Hall, 2003.

[36] J. Duoandikoetxea, *Fourier Analysis*, American Mathematical Society, 2000.

[37] A.N. Kolmogorov, "Sur les fonctions harmoniques conjuguées et les séries de fourier," *Fund. Math.*, vol. 7, pp. 23–28, 1925.

[38] H. Kober, "A note on Hilbert's operator," *Bull. Am. Math. Soc.*, vol. 48, pp. 421–427, 1942.

[39] M. Riesz, "Sur les fonctions conjuguées," *Mathematische Zeitschrift*, pp. 218–244, 1928.

[40] E. Bedrosian, "A product theorem for Hilbert transforms," *Proc. IEEE*, vol. 51, pp. 868–869, 1963.

[41] A.W. Lohmann, D. Mendlovic, and Z. Zalevsky, "Fractional Hilbert transforms," *Optics Letters*, vol. 21, pp. 281–283, 1996.

[42] J.A. Davis, D.E. McNamara, and D.M. Cottrell, "Analysis of the fractional Hilbert transforms," *Applied Optics*, vol. 37, pp. 6911–6913, 1998.

[43] A.I. Zayed, "Hilbert transform associated with the fractional Fourier transforms," *IEEE Signal Process. Lett.*, vol. 5, pp. 206–208, 1998.

[44] R. Tao, X. M. Li, and Y. Wang, "Generalization of the fractional Hilbert transforms," *IEEE Signal Process. Lett.*, vol. 15, pp. 365–368, 2008.

[45] E.M. Stein and G. Weiss, *Introduction to Fourier Analysis on Euclidean Spaces*, Princeton University Press, 1971.

[46] P. Abry and P. Flandrin, "Multiresolution transient detection," *Proc. Int. Symp. on Time-Freq. and Time-Scale Analysis*, pp. 225–228, 1994.

[47] C. Chaux, L. Duval, and J.-C. Pesquet, "Hilbert pairs of M-band orthonormal wavelet bases," *Proc. European Signal and Image Processing Conference*, pp. 6–10, 2004.

[48] T. Olson and J. DeStefano, "Wavelet localization of the Radon transform," *IEEE Transactions on Signal Processing*, vol. 42, pp. 2055–2067, 1994.

[49] K.N. Chaudhury and M. Unser, "Construction of Hilbert transform pairs of wavelet bases and Gabor-like transforms," *IEEE Transactions on Signal Processing*, vol. 57, pp. 3411–3425, 2009.

[50] M. Unser and T. Blu, "Fractional splines and wavelets," *SIAM Review*, vol. 42, no. 1, pp. 43–67, March 2000.

[51] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[52] M. Unser and T. Blu, "Wavelet theory demystified," *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 470–483, February 2003.

[53] J. Ville, "Theorie et application de la notion de signal analytique," *Cables and Transmissions*, vol. 93, no. III, pp. 153–158, 1948.

[54] K. Gröchenig, *Foundations of Time-Frequency Analysis*, Birkhäuser Boston, 2000.

[55] M. Unser and T. Blu, "Construction of fractional spline wavelet bases," *Proc. SPIE Conference on Mathematical Imaging: Wavelet Applications in Signal and Image Processing VII*, vol. 3813, pp. 422–431, July 1999.

[56] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge, 1996.

[57] T. Blu and M. Unser, "The fractional spline wavelet transform: definition and implementation," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 512–515, June 2000.

[58] A. Aldroubi and M. Unser, "Families of multiresolution and wavelet spaces with optimal properties," *Numerical Functional Analysis and Optimization*, vol. 14, pp. 417–446, 1993.

[59] G. Battle, "A bolck spin construction of ondelettes. Part I: Lemarie functions," *Commun. Math. Physics.*, vol. 110, pp. 601–615, 1987.

[60] S. Dubuc, "Symmetric iterative interpolation processes," *Construct. Approx.*, vol. 5, pp. 49–68, 1989.

[61] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, pp. 7–42, 2002.

[62] D. Scharstein and R. Szeliski, "Middlebury Stereo Vision Page (taxonomy and comparison of two-frame stereo correspondence algorithms)," http://vision.middlebury.edu/stereo/.

[63] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[64] Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, pp. 139–154, 1985.

[65] A. Kumar, S. Haker, C. Vogel, A. Tannenbaum, and S. Zucker, "Stereo disparity and L^1 minimization," *Proceedings of the 36th Conference on Decision & Control, San Diego, USA*, pp. 1125–1129, 1997.

[66] T.H. Cormen, E.C. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press & McGraw-Hill, 2001.

[67] D. Geiger, B. Ladendorf, and A.L. Yuille, "Occlusions and binocular stereo," *Proc. European Conference on Computer Vision*, pp. 425–433, 1992.

[68] L. Wang, M. Liao, M. Gong, R. Yang, and D. Nister, "High-quality real-time stereo using adaptive cost aggregation and dynamic programming," *International Symposium on 3D Data Processing Visualization and Transmission*, pp. 798–805, 2006.

[69] S. Mallat, "Zero-crossings of a wavelet transform," *IEEE Transactions on Information Theory*, vol. 41, pp. 3397–3415, 1993.

[70] H-P. Pan, "General stereo matching using symmetric complex wavelets," *Proceedings of SPIE, Wavelet Applications in Signal and Image Processing IV*, vol. 2825, 1996.

[71] L. Di Stefano, M.M. S. Mattoccia, and G. Neri, "A fast area-based stereo matching algorithm," *Image and Vision Computing*, vol. 22, pp. 938–1005, 2004.

[72] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister, "Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 492–504, 2009.

[73] D. Scharstein and R. Szeliski, "Stereo matching with nonlinear diffusion," *International Journal of Computer Vision*, vol. 28, pp. 155–174, 1998.

[74] T. Lindeberg, "Direct estimation of affine image deformations using visual front-end operations with automatic scale selection," *Proc. Fifth International Conference on Computer Vision*, pp. 134–141, 1995.

[75] J. Magarey and N. Kingsbury, "Motion estimation using a complex-valued wavelet transform," *IEEE Transactions on Signal Processing*, vol. 46, pp. 1069–1084, 1998.

[76] J. Magarey and A. Dick, "Multiresolution stereo image matching using complex wavelets," *Proc. Fourteenth International Conference on Pattern Recognition*, vol. 1, pp. 16–20, 1998.

[77] K.N. Chaudhury, A. Muñoz-Barrutia, and M. Unser, "Fast space-variant elliptical filtering using box splines," *IEEE Transactions on Image Processing*, vol. 19, pp. 2290–2306, 2010.

[78] R. Deriche, "Fast algorithms for low-level vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 78–87, 1990.

[79] I.T. Young and L.J. van Vliet, "Recursive implementation of the Gaussian filter," *Signal Processing*, vol. 44, no. 2, pp. 139–151, 1995.

[80] S. Tan, J.L. Dale, and A. Johnston, "Performance of three recursive algorithms for fast space-variant Gaussian filtering," *Real-Time Imaging*, vol. 9, no. 3, pp. 215–228, 2003.

[81] J. Geusebroek, A. Smeulders, and J. van de Weijer, "Fast anisotropic Gaussian filtering," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 938–943, August 2003.

[82] S.Y.M. Lam and B.E. Shi, "Recursive anisotropic 2-D Gaussian filtering based on a triple-axis decomposition," *IEEE Transactions on Image Processing*, vol. 16, no. 7, pp. 1925–1930, 2007.

[83] J.-S. Lee, "Digital image enhancement and noise filtering by use of local statistics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 165–168, 1980.

[84] J. Weickert, "Theoretical foundations of anisotropic diffusion in image processing," *Proc. Theoretical foundations of Computer Vision*, vol. 11, pp. 221–236, 1994.

[85] X. Wang, "On the gradient inverse weighted filters," *IEEE Transactions on Signal Processing*, vol. 40, no. 2, pp. 482–484, 1992.

[86] A. Muñoz-Barrutia, R. Ertlé, and M. Unser, "Continuous wavelet transform with arbitrary scales and $O(N)$ complexity," *Signal Processing*, vol. 82, no. 5, pp. 749–757, May 2002.

[87] A. Muñoz-Barrutia, X. Artaechevarria, and C. Ortiz-de Solorzano, "Spatially variant convolution with scaled B-splines," *IEEE Transactions on Image Processing*, vol. 19, pp. 11–24, 2010.

[88] M. Richter, "Use of box splines in computer tomography," *Computing*, vol. 61, no. 2, pp. 133–150, 1998.

[89] T. Asahi, K. Ichige, and R. Ishii, "An efficient algorithm for decomposition and reconstruction of images by box splines," *IEICE Transactions on Fundamentals of Electronics, Commun. and Comp. Sc.*, vol. E 84, no. 8, pp. 1883–1891, 2001.

[90] L. Condat and D. Van De Ville, "Three-directional box-splines: Characterization and efficient evaluation," *IEEE Signal Process. Lett.*, vol. 13, no. 7, pp. 417–420, 2006.

[91] A. Entezari, D. Van De Ville, and T. Möller, "Practical box splines for reconstruction on the body centered cubic lattice," *IEEE Transactions Vis. Comput. Graphics*, vol. 14, no. 2, pp. 313–328, 2008.

[92] P.B. Zwart, "Multivariate splines with nondegenerate partitions," *SIAM Journal on Numerical Analysis*, vol. 10, pp. 665–673, 1973.

[93] A. Tkacenko and P.P. Vaidyanathan, "Generalized kurtosis and applications in blind equalization of MIMO channels," *Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 742–746, 2001.

[94] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[95] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling and Simulation*, vol. 4, pp. 490–530, 2005.

[96] B. Jähne, *Digital Image Processing*, Springer, 1997.

[97] A. Buades, B. Coll, and J.M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling and Simulation*, vol. 4, pp. 490–530, 2005.

[98] W.C.K. Wong, A.C.S. Chung, and S.C.H. Yu, "Trilateral filtering for biomedical images," *IEEE International Symposium on Biomedical Imaging*, 2004.

[99] E.P. Bennett, J.L. Mason, and L. McMillan, "Multispectral bilateral video fusion," *IEEE Transactions on Image Processing*, vol. 16, pp. 1185–1194, 2007.

[100] E.P. Bennett and L. McMillan, "Video enhancement using per-pixel virtual exposures," *Siggraph*, vol. 24, pp. 845–852, 2005.

[101] B.M. Oh, M. Chen, J. Dorsey, and F. Durand, "Image-based modeling and photo editing," *Siggraph*, pp. 433–442, 2001.

[102] R. Ramanath and W. E. Snyder, "Adaptive demosaicking," *Journal of Electronic Imaging*, vol. 12, pp. 633–642, 2003.

[103] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi, "Bilateral filtering-based optical flow estimation with occlusion detection," *European Conference on Computer Vision*, pp. 211–224, 2006.

[104] P. Sand and S. Teller, "Particle video: Long-range motion estimation using point trajectories," *International Journal of Computer Vision*, vol. 80(1), pp. 72–91, 2006.

[105] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister, "Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling," *IEEE Transaction on Pattern Analyiss and Machine Intelligence*, vol. 31, pp. 492–504, 2009.

[106] K.J. Yoon and I.S. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 650–656, 2006.

[107] F. Porikli, "Constant time $O(1)$ bilateral filtering," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[108] Q. Yang, K.-H. Tan, and N. Ahuja, "Real-time $O(1)$ bilateral filtering," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 557–564, 2009.

[109] F. C. Crow, "Summed-area tables for texture mapping," *ACM Siggraph*, vol. 18, pp. 207–212, 1984.

[110] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Siggraph*, vol. 21, pp. 257–266, 2002.

[111] K.N. Chaudhury, Zs. Püspöki, A. Muñoz Barrutia, D. Sage, and M. Unser, "Fast detection of cells using a continuously scalable Mexican-hat-like template," *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1277–1280, 2010.

[112] C. Vonesch, F. Aguet, J.-L. Vonesch, and M. Unser, "The colored revolution of bioimaging," *IEEE Signal Processing Mag.*, vol. 23(3), pp. 20–31, 2006.

[113] R. G. Prabhakar, K. Nandy, J. Collins, K. J. Meaburn, T. Misteli, and S. J. Lockett, "A high-throughput system for segmenting nuclei using multiscale techniques," *Cytometry*, vol. 73A(5), pp. 451–466, 2008.

[114] N. Malpica, C. Ortiz-de Solorzano, J.J. Vaquero, A. Santos, I. Vallcorba, J.M. García-Sagredo, and F. del Pozo, "Applying watershed algorithms to the segmentation of clustered nuclei," *Cytometry*, vol. 28(4), pp. 289–297, 1997.

[115] C. Ortiz-de Solorzano, R. Malladi, S.A. Lelievre, and S.J. Lockett, "Segmentation of nuclei and cells using membrane related protein markers," *Journal of Microscopy*, vol. 201(3), pp. 404–415, 2001.

[116] J. Deng and H.T. Tsui, "A fast level set method for segmentation of low contrast noisy biomedical images," *Pattern Recognition Letters*, vol. 23, pp. 161–169, 2002.

[117] J. Byun, M.R. Verardo, B. Sumengen, G.P. Lewis, B.S. Manjunath, and S.K. Fisher, "Automated tool for the detection of cell nuclei in digital microscopic images: Application to retinal images," *Molecular Vision*, vol. 12, pp. 949–960, 2006.

[118] X. Zhou and Wong S.T.C., "High content cellular imaging for drug development," *IEEE Signal Processing Mag.*, vol. 23(2), pp. 170–174, 2006.

[119] J. Kiefer, "Sequential minimax search for a maximum," *AMS Proceedings*, vol. 4, pp. 502–506, 1953.

[120] M.A. Miller and N.G. Kingsbury, "Image denoising using derotated complex wavelet coefficients," *IEEE Transactions on Image Processing*, vol. 17, pp. 1500–1511, 2008.

[121] Y. Zhang and N.G. Kingsbury, "Restoration of images and 3D data to higher resolution by deconvolution with sparsity regularization," *IEEE Conference on Image Processing*, 2010.

[122] P. F. C. de Rivaz and N. G. Kingsbury, "Complex wavelet features for fast texture image retrieval," *Proc. IEEE Conf. on Image Processing*, 1999.

[123] R.R. Coifman, Y. Meyer, and M. Wickerhauser, "Wavelet analysis and signal processing," *Wavelets and theirApplications*, pp. 153–178, 1992.

Curriculum Vitæ

Biomedical Imaging Group
École Polytechnique Fédérale de Lausanne
BM 4.134, Station - 17
CH-1015 Lausanne VD
Switzerland

http://bigwww.epfl.ch/chaudhury
k.n.chaudhury@ieee.org
kunal.chaudhury@epfl.ch
Tel: +41 21 693 5143
Fax: +41 21 693 6810

PERSONAL DATA

Date of Birth	28th August, 1980
Place of Birth	Siliguri, India
Nationality	Indian

EDUCATION

2006 – 2011	Ph. D.	Computer, Communication and Information Science	École Polytechnique Fédérale de Lausanne Switzerland
2003 – 2005	M. E.	System Science and Automation	Indian Institute of Science Bangalore, India
1999 – 2003	B. E.	Electrical Engineering	Jadavpur University Kolkata, India

RESEARCH INTERESTS

Time-frequency and time-scale analysis, Computational harmonic analysis, Digital image processing, Inverse problems in imaging.

AWARDS

- Fellowship awarded by the Swiss National Science Foundation (SNSF) for post-doctoral research at Princeton University (2011–2012).

- Best student award (gold medal recipient) for the Masters program in Systems Science and Automation, Department of Electrical Engineering, Indian Institute of Science (2005).

- Scholarship awarded by the Ministry of Human Resources Development (MHRD), Govt. of India (2003-2005).

- All India Rank of 19 (99.9 percentile) in Graduates Aptitude Test Examination (GATE) in Electrical Engineering for graduate admissions in India (2003).

- Stood 2nd in the Maths Olympiad organized at the city level (1996).

PROFESSIONAL ACTIVITIES

Member	Institute of Electrical and Electronics Engineers (IEEE)
	Society for Industrial and Applied Mathematics (SIAM)
Reviewer	IEEE Transaction on Image Processing
	IEEE Transaction on Signal Processing
	Signal Processing
	IEEE Transaction on Circuits and Systems
	IEEE Signal Processing Letters
	Pattern Recognition Letters

TEACHING

- Teaching Assistant at École Polytechnique Fédérale de Lausanne (EPFL) for the Master's level courses on Image Processing between 2006-2010.

- Supervised various student projects at EPFL.

PROFESSIONAL EXPERIENCE

Employed as a Research Engineer jointly at Siemens Corporate Technology (SCT), and Siemens Informations Systems Limited (SISL), Bangalore, India (2005-2006).

TECHNICAL SKILLS

- Programming Languages: C, C++, Java.
- Packages: MATLAB, Mathematica.
- Operating Systems: Macintosh OS X, Unix,Windows.

PUBLICATIONS

Journal Articles

1. **K.N. Chaudhury**, Daniel Sage, and M. Unser, "**Fast $O(1)$ bilateral filtering using trigonometric range kernels**", IEEE Transactions on Image Processing, submitted.

2. **K.N. Chaudhury**, and M. Unser, "**On the Hilbert transform of wavelets**", IEEE Transactions on Signal Processing, in press.

3. **K.N. Chaudhury**, A. Munoz-Barrutia, and M. Unser, "**Fast space-variant elliptical filtering using box splines**," IEEE Transactions on Image Processing, vol. 19, pp. 2290–2306, 2010.

4. **K.N. Chaudhury**, and M. Unser, "**On the shiftability of dual-tree complex wavelet transforms**," IEEE Transactions on Signal Processing, vol. 58, pp. 221–232, 2010.

5. **K.N. Chaudhury**, and M. Unser, "**Construction of Hilbert transform pairs of wavelet bases and Gabor-like transforms**," IEEE Transactions on Signal Processing, vol. 57, pp. 3411–3425, 2009.

6. **K.N. Chaudhury**, and K.R. Ramakrishnan, "**Stability and convergence of the level set method in computer vision**," Pattern Recognition Letters, vol. 28, pp. 884-893, 2007.

Conference Proceedings

1. **K.N. Chaudhury**, Zs. Puspoki, A. Munoz-Barrutia, D. Sage, and M. Unser, "**Fast detection of cells using a continuously scalable Mexican-hat-like template**," Proc. IEEE International Symposium on Biomedical Imaging (ISBI), Rotterdam, The Netherlands, pp. 1277-1280, 2010.

2. **K.N. Chaudhury**, and M. Unser, "**Gabor wavelet analysis and the fractional Hilbert transform**," Proc. SPIE Optics and Photonics 2009 Conference on Mathematical Methods: Wavelet XIII, San Diego CA, USA, vol. 7446, pp. 74460T-1/74460T-7, 2009.

3. **K.N. Chaudhury**, and M. Unser, "**The fractional Hilbert transform and dual-tree Gabor-like wavelet Analysis**," Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Taipei, Taiwan, pp. 3205-3208, 2009.

4. **K.N. Chaudhury**, A. Munoz-Barrutia, and M. Unser, "**Fast adaptive elliptical filtering using box splines**," Proc. IEEE International Conference on Image Processing (ICIP), San Diego CA, USA, pp. 785-788, 2008.

5. **K.N. Chaudhury**, and M. Unser, "**Construction of Hilbert transform pairs of wavelet bases and optimal time-frequency localization**," Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Las Vegas NV, USA, pp. 3277-3280, 2008.