NONDYADIC AND NONLINEAR MULTIRESOLUTION IMAGE APPROXIMATIONS

THÈSE N^{O} 2555 (2002)

PRÉSENTÉE À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES TECHNIQUES

PAR

Maria Arrate MUÑOZ BARRUTIA

Ingénieure en télécommunications, Universidad Pública de Navarra, Espagne
et de nationalité espagnole

acceptée sur proposition du jury:

Prof. M. Unser, directeur de thèse
Prof. M. Barlaud, rapporteur
Prof. M. Vetterli, rapporteur
Prof. S. Voloshynovskiy, rapporteur

Lausanne, EPFL
2002

Nondyadic and Nonlinear Multiresolution Image Approximations

Arrate Muñoz Barrutia

Thèse N° 2555 (2002)

Thèse présentée au faculte de sciences et techniques de l'ingénieur
pour l'obtention du grade de docteur ès sciences techniques
et acceptée sur proposition du jury:

Prof. René Salathé, *président*
Prof. Michael Unser, *directeur de thèse*
Prof. Martin Vetterli, *co-rapporteur*
Prof. Sviatoslav Voloshynovkiy, *co-rapporteur*
Prof. Michel Barlaud, *co-rapporteur*

Ecole Polytechnique Fédérale de Lausanne—2002

Cover design by Annette Unser
Cover printed by Copy-Tunnel
Printing and binding by Repro-EPFL
Typeset with LaTeX

Available at `http://bigwww.epfl.ch/munoz/`

Abstract

This thesis focuses on the development of novel multiresolution image approximations. Specifically, we present two kinds of generalization of multiresolution techniques: image reduction for arbitrary scales, and nonlinear approximations using other metrics than the standard Euclidean one.

Traditional multiresolution decompositions are restricted to dyadic scales. As first contribution of this thesis, we develop a method that goes beyond this restriction and that is well suited to arbitrary scale-change computations. The key component is a new and numerically exact algorithm for computing inner products between a continuously defined signal and B-splines of any order and of arbitrary sizes. The technique can also be applied for non-uniform to uniform grid conversion, which is another approximation problem where our method excels. Main applications are resampling and signal reconstruction.

Although simple to implement, least-squares approximations lead to artifacts that could be reduced if nonlinear methods would be used instead. The second contribution of the thesis is the development of nonlinear spline pyramids that are optimal for ℓ_p-norms. First, we introduce a Banach-space formulation of the problem and show that the solution is well defined. Second, we compute the ℓ_p-approximation thanks to an iterative optimization algorithm based on digital filtering. We conclude that ℓ_1-approximations reduce the artifacts that are inherent to least-squares methods; in particular, edge blurring and ringing. In addition, we observe that the error of ℓ_1-approximations is sparser. Finally, we derive an exact formula for the asymptotic L_p-error; this result justifies using the least-squares approximation as initial solution for the iterative optimization algorithm when the degree of the spline is even; otherwise, one has to include an appropriate correction term.

The theoretical background of the thesis includes the modelisation of images in a continuous/discrete formalism and takes advantage of the approximation theory of linear shift-invariant operators. We have chosen B-splines as basis

functions because of their nice properties. We also propose a new graphical formalism that links B-splines, finite differences, differential operators, and arbitrary scale changes.

Résumé

Le sujet de cette thèse est le développement de nouvelles approximations multiéchelle d'image. Plus précisément, nous présentons deux types de généralisation de techniques multiéchelles: réduction d'image par un facteur arbitraire, et approximations non-linéaires basées sur des métriques qui diffèrent de la métrique euclidienne habituelle.

Les décompositions multiéchelle traditionnelles se limitent aux échelles dyadiques. Comme première contribution de cette thèse, nous développons une méthode qui surmonte cette restriction et qui s'adapte bien aux calculs de changement arbitraire d'échelle. L'élément clef en est un nouvel algorithme, numériquement exact, pour le calcul du produit vectoriel entre un signal défini continûment et des B-splines de tout ordre et de taille arbitraire. Cette technique peut aussi s'appliquer à la conversion d'une grille non-uniforme en une grille uniforme, qui est un autre exemple d'approximation où notre méthode est particulièrement performante. Les applications principales sont le ré-échantillonnage et la reconstruction de signal.

Bien qu'elles soient simples à implémenter, les approximations au sens des moindres carrés conduisent à des défauts qui pourraient être réduits si elles étaient remplacées par des méthodes non-linéaires. La seconde contribution de cette thèse est le développement de pyramides non-linéaires, basées sur des splines, et optimales en normes ℓ_p. Tout d'abord, nous formulons le problème dans un espace de Banach et nous montrons que la solution est bien définie. Ensuite, nous calculons l'approximation au sens ℓ_p grâce à un algorithme itératif d'optimisation construit à l'aide de filtres numériques. Nous concluons que les approximations au sens ℓ_1 réduisent les défauts liés aux méthodes des moindres carrés; en particulier, perte de netteté des contours et tintements. De plus, nous observons que l'erreur des approximations au sens ℓ_1 est moins dense. Enfin, nous déduisons une formule exacte pour l'erreur asymptotique au sens L_p; ce résultat justifie de l'usage d'une approximation au sens des moindres carrés

comme solution initiale de l'algorithme itératif d'optimisation.

Le contenu théorique de cette thèse inclut la modélisation d'images dans un formalisme continu/discret et tire parti de la théorie d'approximation d'opérateurs linéaires à invariance de phase. Nous avons choisi des B-splines comme fonctions de base en raison de leur propriétés élégantes. Nous proposons en outre un nouveau formalisme graphique qui relie B-splines, différences finies, opérateurs différentiels, et changements arbitraires d'échelle.

To my parents.

Acknowledgements

My greatest thanks go to my advisor Michael Unser. His guidance made this thesis a reality. I really appreciate his patience and scientific encouragements. The people at the Biomedical Imaging Group (BIG)[1] create a very supportive and helpful environment. Thanks to Thierry Blu for his enormous scientific support and to share his sensitivity for science, friendship, and art. Thanks also to Philippe Thévenaz for inculcating discipline and helping to writing up documents. The time and effort he put in reviewing the drafts of this thesis need to be specially acknowledged. I would also like to thank Daniel Sage, for his dedication to give an excellent computer support. My office mate Jan Kybic has been wonderful: I miss him!. It has been very pleasant to share time with Slavica Jonić and the rest of the BIG people. The discussions with Akram Aldroubi during the SPIE'00 conference were very productive. Many thanks to my friends in Lausanne, with whom I had a great time. The atmosphere in the Doctoral School was specially close and supportive. My deepest gratitude goes to María José Carrillo whom I will always consider as part of my family. The happiness of the Spanish and Italian community have given to me a lot of positive energy during all these years. I am also very grateful to my old friends in Pamplona and to those I met in London who are always present to share important moments. Special thanks to Iñako for his love, strength, and understanding. I would like to dedicate this work to my family, in particular to my parents, for their love and encouragement.

Arrate
Lausanne, February 2002

[1] The work on this thesis was done in collaboration with and supervised by Michael Unser and Thierry Blu.

Contents

Chapter 1

Introduction

1.1 Contributions of the thesis

This thesis focuses on the development of novel multiresolution image approximations. Our goals are twofold:

- Extend multiresolution techniques to arbitrary scales.
- Construct non-linear multiresolution schemes that are artifact-free and more robust than their least-squares counterparts.

The theoretical background of the thesis includes the modeling of images in a continuous/discrete formalism and takes advantage of the approximation theory of linear shift-invariant operators.

1.2 Motivation

1.2.1 On the limitations of the discrete representation

Figure 1.1 shows a construction done with Lego bricks. These bricks have several colors (here: dark and light). Together, the disposition of bricks forms a nice pattern that we interpret as writing the word "Muñoz". By design, Lego bricks can occupy only specific locations on the plastic substrate; it is not possible to set a brick halfway between two sites. When such a situation occurs, that is, when only a limited set of locations is available, the system is called discrete. A

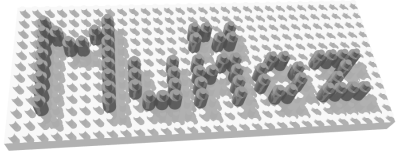


Figure 1.1: Discrete image constructed with Lego bricks.

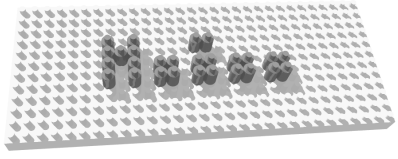


Figure 1.2: Can you guess the word?.

typical setting that produces discrete data is a digital camera, where each pixel contains a number that codes the intensity of light impinging on the sensor at that particular location.

A common problem to digital cameras and to the Lego world of Figure 1.1 is that of resizing. Suppose we want to reduce the size of the text in Figure 1.1. Because there are only discrete admissible sites, it is difficult to come up with a solution that would be still easily readable, but that would take, say, half the size. Figure 1.2 gives an example of a failed attempt. Similarly, it is not trivial to resize a picture once it is discretized. In general, this problem is pervasive in many areas of digital signal processing, for example in the context of numerical printers and photocopy machines, and applies to reduction and magnification as well.

The standard solution to this problem is based on the introduction of a continuous image model (interpolation) followed by a resampling step. In this thesis, we use the same type of interpolation model, but we will formulate the resizing as a minimization of a cost function.

1.2.2 What is multiresolution?

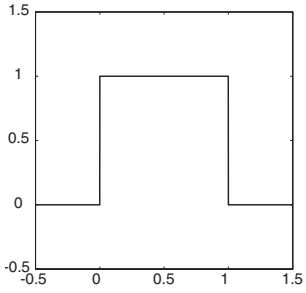


Figure 1.3: Haar scaling function.

When performing image magnification, the number of pixels that represent the digital image increases. Nevertheless, the information content of the resulting image remains nearly the same. The dual operation is image reduction which results into a decrease of the number of pixels. Consequently, we lose some amount of information, the magnitude of which depends on the image content, on the value of the scale, and on the method applied for the computation.

All digital images, in particular those we are able to store and process in a computer, are discrete. To construct a continuous version out of a discrete image, we make use of the box function represented in Figure 1.3, which is commonly called the Haar scaling function, and of all its integer translates. We apply it to both columns and rows, in separable fashion. The process of fitting a continuous model to a discrete image is called interpolation. In the present example, the resulting image is a bar graph—it is locally constant, with possible discontinuities at the integers. This interpolation model is the simplest one may think of; more sophisticated ones will be considered in this thesis. Figure 1.4 illustrates the effect of reducing by successive powers of two the image of cells shown on top. The left column corresponds to the progressive reduction of images. The crudest way to achieve a reduction by a factor two—to represent the function with only a quarter of the bars—is to throw three bars out of four. This leads to a large approximation error. Obviously, we can reduce it if we substitute each block of four bars by their average value. In the present example, the squared approximation error would reach its minimum (least-squares approximation).

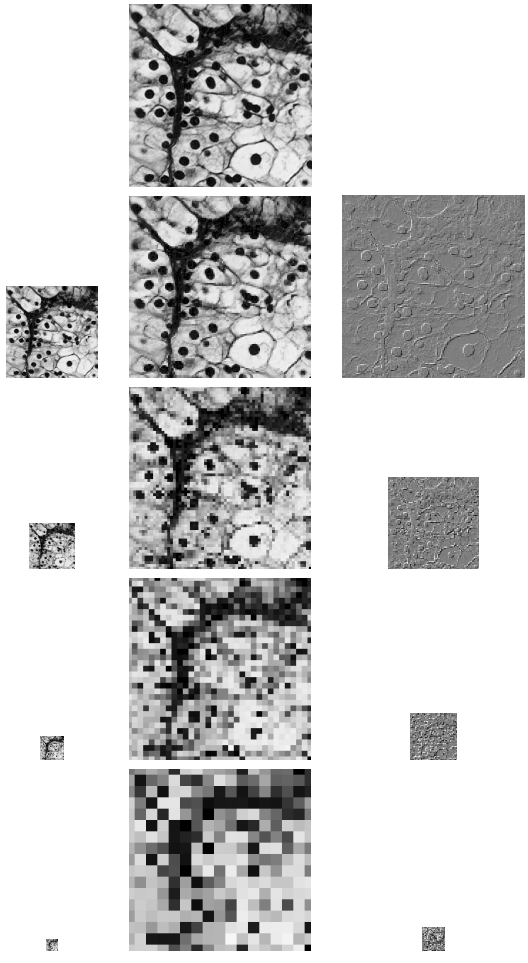


Figure 1.4: Multiresolution decomposition of the image of cells shown on top. The scale reduction from one level to the next is two (from one level to the next, the total number of pixels gets divided by four, that is, by two in each dimension). Left column: Reduced images. Center column: Version of the coarse images that is expanded back to the original size. Right column: Difference between the image in the left column expanded by a factor two and the one above it.

We can evaluate qualitatively the amount of information lost at each step by expanding back the reduced images to the original size and by examining them visually. The result of this operation corresponds to the images shown in the center column. We observe that the cell membranes disappear early. For larger reductions, it becomes more and more difficult to perceive the nuclei of the cells. At the coarsest level, the dark wall separating cells, that was clearly visible in the images with moderate reduction, is now barely discernible; it is the last and coarsest feature left. Another striking element is the blocking appearance of the images which is due to the use of a piecewise constant model for the interpolation.

At each resolution level, the size of the square blocks is proportional to the number of pixels of the original image that are transformed into a single pixel in the reduced images. The right column corresponds to the difference between the image in the left column expanded by a factor two and the one above it. This difference is a quantitative measure of the amount of information lost at each reduction step. To recover an image with more details, it is sufficient to add the difference image expanded back to the original size to a coarse image; by proceeding recursively, one can recover the initial image (perfect reconstruction condition).

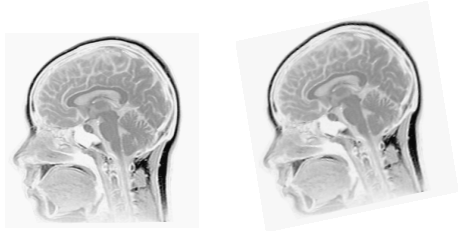
From this example we can easily conclude that natural images contain information not at one but at several scales or resolutions.

The membranes, the nuclei, and the dark wall for example, are structures of different size. When performing the reduction, we are losing at each level details of a size comparable to that of the factor of reduction. Consequently, we need several resolutions to analyze an image. Multiple-resolution descriptions of the type shown in Figure 1.4 are called multiresolution, or hierarchical descriptions [62].

1.2.3 Why multiresolution is important?

Multiresolution image decompositions have many applications in image processing. Figure 1.5 gives an example of image registration [78]. The two MRI images at the top belong to the same patient and were taken a few months apart. The radiologist needs to compare them to detect or quantify the changes that might have taken place in-between acquisitions. To be able to do so, he must align the images; in other words, a geometric correspondence function must be found to warp one of them. It is evident that a registration algorithm working at coarse resolution (fewer number of pixels) will speed up computations and will converge more easily since the contribution of noise and details is reduced. The

rough solution for the correspondence function obtained at a given level of the pyramid can be propagated to higher-resolution images, where only a slight additional nudge is all what is needed to perfect the registration.



Reference. Original.

Registration algorithm working at coarse resolution.

Figure 1.5: Example of a multiscale registration algorithm. Top: Reference to which an original image must be aligned. Bottom: Illustration of a multiscale registration method that starts by aligning only the large-scale features of the data at a coarse-resolution level. At finer levels, only small corrections for progressively finer details are necessary.

Multiscale processing has numerous applications in image processing and computer vision. Some examples related to the biomedical field are: image registration and warping [71, 92, 121], reconstruction of the electrical activity of the brain from electroencephalogram and magnetoencephalogram signals [46], and snake algorithms for the detection of parametric contours [25].

From another point of view, least-squares pyramidal decompositions provide

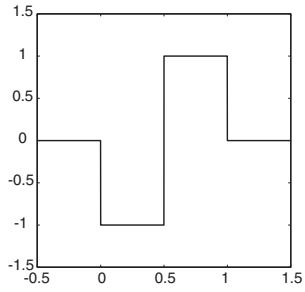


Figure 1.6: Detail Haar function (Haar wavelet).

a sparse or efficient representation of images. A representation being sparse means that the essential information is captured in just a few components. In other words, the error images are composed mainly of very small values and only few large ones. This key property is the basis for the success of multiresolution decompositions in coding and denoising applications [79, 114, 135].

A nice point in favor of these multiresolution constructions is that physiological experiments found analogies between multiresolution schemes and the hierarchical organization of the human primary visual cortex [63]. The human visual system takes advantage of very efficient (sparse) representations of the visual input to speed up the process of interpreting and modelling the information. In addition, this may achieve a minimization of the energy consumption since the essential information is captured with a minimum number of excited neurons [47].

1.2.4 On the limitations of standard multiresolution techniques

Dyadic scales

Restricting the admissible reduction factors to powers of two is typical of a special type of multiresolution analysis: the wavelet transform. Wavelets are basis functions that provide an efficient encoding of the difference of images between two resolutions. The wavelet that corresponds to the Haar example of Figure 1.4 is shown in Figure 1.6 [79]. Note that this Haar wavelet can be easily constructed by a combination of two Haar functions that involves shifting, negating and scaling operations on the function shown in Figure 1.3. Wavelet

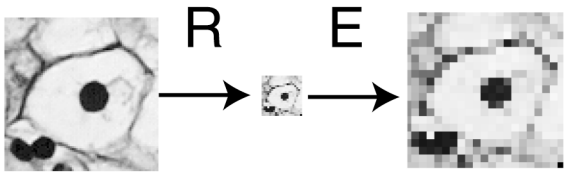


Figure 1.7: Experimental set-up: R stands for reduction. E stands for expansion.

transforms are least-squares approximations that ensure perfect reconstruction. They are non-redundant approximations that are represented with the same number of pixels than the original signal. They can be computed with fast algorithms, which is one of the main reason why the use of wavelet transforms is widespread.

For applications such as posting a picture on the Web, we would like to be able to choose its size freely, while keeping a reasonable quality, for example like that given by a least-squares approximation. In the thesis, we propose such a least-squares method to resize an image by an arbitrary scale factor. Its implementation is easy as the algorithm only requires digital filtering and matrix multiplications.

Artifacts

We will now review the artifacts that affect common linear approximations: blocking, blurring, and ringing [120]. To illustrate them, we have performed the experiment shown schematically in Figure 1.7: First, we select a cell which is part of Figure 1.4-top. Second, we produce a reduced version using one of the numerous linear approximation methods available. Third, we expand back the reduced image to recover its initial size. Finally, we compare it with the original. We have done the experiment for two different scale factors: four, and eight.

Blocking We have already encountered in Figure 1.4 an example of the use of a piecewise-constant interpolation model. Its advantage is to be the simplest model one can think of, and one of its disadvantages is that it produces a pixelized appearance upon magnification. For example, when the scale factor is two, whole blocks of four bars in the magnified image depend on just a single

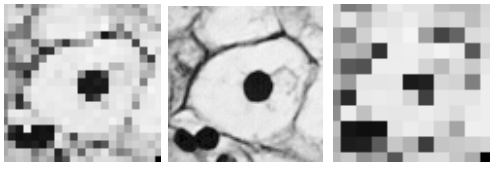


Figure 1.8: Blocking of piecewise constant model. Left: Reduction/Expansion by four. Center: Original. Right: Reduction/Expansion by eight. The pixelized appearance of the image is called blocking which also results in shape deformations.

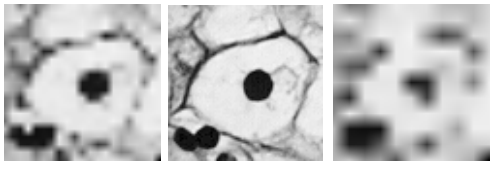


Figure 1.9: Blur of bilinear model. Left: Reduction/Expansion by four. Center: Original. Right: Reduction/Expansion by eight. The blur gives an out-of-focus appearance to the resulting image.

bar of the image to magnify. This mechanism results in the square blocks that are apparent in Figure 1.8. This defect also deforms the shape of the structures.

Blurring The blocking effect disappears when we use a more complicated model for the interpolation, for example, piecewise linear. This is shown in Figure 1.9, where the smooth regions of the image are reasonably well approximated. Nevertheless, when performing reduction, the edges will get heavily smoothed, which gives an out-of-focus appearance to the resulting image. Moreover, a more detailed analysis reveals some distortion and some folding in the frequency domain; this phenomenon is called aliasing and was also present in the case of the piecewise-constant interpolation model.

Ringing The approximation methods that use very simple interpolation models, such as those illustrated in Figures 1.8 and 1.9, have the advantage of creating no spurious oscillations. Thus, they are adequate to handle smooth

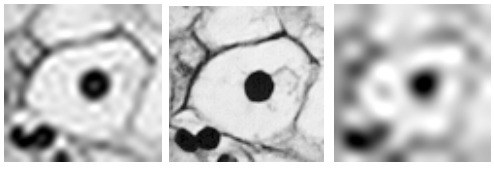


Figure 1.10: Ringing. Left: Reduction/Expansion by four. Center: Original. Right: Reduction/Expansion by eight. The oscillations caused by ringing result in the creation of an incorrect spot of lighter intensity in the center of the cell nucleus of the left image and a texturized appearance of both-left and right-images.

regions. Unfortunately, in the regions where the image is insufficiently smooth, this type of approximations is unsuccessful and may be significantly improved by using higher-order polynomial models. In turn, these better approximations have the inconvenient of introducing Gibbs oscillations [79], because they are not capable of correctly representing discontinuities (i.e., the contours). These oscillations are visually disturbing as shown in Figure 1.10.

The types of artifact we just described generally reduce the performance and limit the range of applications of approximation methods. The speed of convergence and the robustness of multiscale algorithms is also negatively affected.

When calculating a coarse-resolution approximation, we would like to keep in the coarse resolution the largest possible amount of descriptive information and to avoid creating artifacts. It follows that most entries in the corresponding detail—or error—image should vanish (i.e., the detail image should be very sparse). To achieve this, we would tend to prefer high-order polynomial interpolation models (i.e., cubic degree), while taking into account the compromise between reduction of the blurring and the computational complexity. Unfortunately, the ringing effect is contaminating these types of approximation. As this artifact is exacerbated by the least-squares method, we propose in this thesis to minimize other measures of the error that are more appropriate to preserve shapes and to reduce oscillations.

A typical example of the type of improvement that can be obtained with our nonlinear approximations is shown in Figure 1.11. When comparing Figure 1.10 and 1.11, we can appreciate that the images in the latter case do not present spurious textures, even though the image model (cubic spline) is the same in both cases. In this thesis, we explain in detail how to obtain the much better

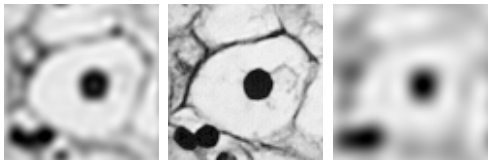


Figure 1.11: Our nonlinear method. Left: Reduction/Expansion by four. Center: Original. Right: Reduction/Expansion by eight. The improvement in the quality of approximation by our nonlinear method is impressive.

result of Figure 1.11.

1.3 Related work

Mallat's multiresolution theory makes the connection between the wavelet transform in harmonic analysis, and filterbanks in discrete signal processing [76, 84]. Such a connection allows the wavelet transform, initially defined in the continuous domain, to be computed with fast algorithms based on filterbanks [114]. Nevertheless, wavelet multiresolution theory is limited to scale changes that are powers of two. Blu [17] and Kovačević *et. al* [69] have shown that rational scale changes were possible by relaxing the shift-invariant condition on the multiresolution theory. The drawback is a very complicated filter design.

Another tool to construct multiresolution image decompositions are pyramids. The pyramid scheme was introduced by Burt and Adelson [27] for coding applications. Classically, both types of scheme (wavelets and pyramids) are implemented with linear filters. In the recent years, the use of nonlinear filters has become commonplace to treat some problems related to images processing, whenever linear approaches are not really appropriated. The most relevant families into which nonlinear filters can be classified are median filters [14], morphological filters [54, 109], and order-statistic filters [10, 90, 91].

The pyramid scheme seems to be specially well-suited for extending linear methods into nonlinear ones. Redundancy gives flexibility in the choice of the filters which can be nonlinear operators [135]. For example, techniques that construct nonlinear multiresolution approximations can be obtained by replacing linear filters by median [12, 42, 81, 83, 111], morphological operators [31, 51, 95, 122], and rank-order filters [88].

For applications such as coding, the class of nonredundant (or maximally-decimated) decompositions are preferred. Therefore, some extensions of this class to nonlinear filter banks have been considered. They often follow novel design and implementation methods which were initially designed for linear subband decompositions, such as the ladder structure [26] and, more recently, the lifting scheme [52, 115, 116]. From the literature of critically decimated nonlinear filterbanks, we can cite useful methods such as morphological subband decompositions and filterbanks [36, 45, 53], order-statistics-based subband decompositions and filterbanks [11, 15, 104], and morphological wavelet decompositions [32, 43, 44, 51, 55].

1.4 Thesis outline

The thesis is organized as follows: In Chapter 2, we review some concepts of approximation theory which are fundamental tools for the next chapters; in particular, we examine the question of characterizing the approximation error as a function of the sampling step. We give a mathematical description of our continuous/discrete image model and of the interpolation and approximation operators.

In Chapter 3, we establish a catalog of mathematical operations; by presenting them in a graphical form, we aim at easing the design of our approximation algorithms. We also present B-splines as the basis functions that we consider most appropriate for developping our continuous/discrete formulation. We present a novel and numerically exact algorithm for computing inner products between a continuously-defined signal and B-splines of any order and of arbitrary sizes.

The next chapters are devoted to practical applications of the theoretical methods presented in the previous two chapters. In Chapter 4, we generalize the spline-resizing algorithm of [73] to splines of any order. In Chapter 5, we develop an original least-squares nonuniform to uniform grid-conversion algorithm.

In Chapter 6, we introduce spline pyramids that are optimal for ℓ_p-norms as an alternative to the traditional multiresolution linear methods. We present a Banach-space formulation of the problem and show that its solution is well-defined. Then, we develop an iterative algorithm based on digital filtering. To conclude, we show that the corresponding reduction operator for values of p close to 1 reduces the blurring and the ringing artifacts inherent to linear pyramids. We also characterize quantitatively the increase in sparsity of ℓ_1 over ℓ_2 approximations.

The main contribution of Chapter 7 is to characterize the error of the L_p-approximation when the sampling step tends to zero. This result justifies that the least-squares approximation be used as initial solution for the iterative optimization algorithm when the degree of the spline is even; otherwise, one has to include an appropriate correction term.

The appendix is dedicated to another application of the mathematical formulation presented in Chapter 3: the continuous wavelet transform. We develop a fast algorithm based on a look-up table to speed up the computations. Finally, we demonstrate the convenience of using this algorithm in the analysis of bowel movements.

Chapter 2

Sampling and Linear Approximation in $L_2(\mathbb{R})$

In some areas of applied mathematics (i.e., finite elements), the problems formulated in the continuous domain are commonly solved using numerical methods. In that case, it is necessary to find a reasonable discrete representation of the initial continuous function. In digital signal processing (DSP), most of the research is made with data already discretized. Here, the concern is to reconstruct a continuous function from its uniformly taken samples which is an unavoidable step when continuously-defined operators such as derivatives need to be implemented.

In approximation theory, a general approach to approximate is to project the continuous function onto a linear and shift-invariant space. An important parameter to describe the space is its scale which determines the quality of the approximation. One of the simplest form of projection is interpolation. This method enssures that the value of the initial function and its approximated version are the same at the sample positions. It is important to characterize the approximation error since the adequacy of the fit between continuous and discrete data needs to be controlled.

In this chapter, we will review the approximation properties of linear operators that approximate continuous functions into some scaled shift-invariant subspace. We are specially interested in the polynomial preserving operators interpolative and projective. In addition, we will review how to characterize the approximation error in function of the scale parameter when the function

to approximate is sufficiently regular.

2.1 Approximation space

The main goal of approximation theory is to find a reasonable approximation at scale T which we denote $f_T(x)$, of a finite energy function $f(x) \in L_2$. A widely used approach considers $f_T(x)$ belonging to the class of functions generated from the integer translates of an appropriate template.

The resulting approximation space is T-integer shift-invariant and can be defined as

$$V_T = \operatorname*{span}_{k\in\mathbb{Z}} \left\{ \varphi\left(\frac{x}{T} - k\right) \right\} \cap L_2, \tag{2.1}$$

where $\varphi(x)$ is some specified basis function with acceptable space and frequency localization.

The function $f(x)$ is approximated at scale T by $f_T(x)$ which is defined as

$$f_T(x) = \sum_{k\in\mathbb{Z}} c_k \varphi\left(\frac{x}{T} - k\right), \tag{2.2}$$

where the c_k's are coefficients that depends linearly on $f(x)$ and specify the approximation model. A discussion on the different methodologies to calculate the value of the coefficients c_k is the object of the next section.

We require that V_T is a closed subspace of L_2 and that each of its functions $f_T(x) \in V_T$ have a unique and stable representation in terms of the coefficients c_k. These conditions are satisfied if $\{\varphi(x-k)\}_{k\in\mathbb{Z}}$ is a Riesz basis of V_1. Mathematically, this means that there exist two constants $0 < A, B < \infty$ such that

$$\forall c \in \ell_2,\ A \cdot \|c\|_{\ell_2} \leq \left\| \sum_{l\in\mathbb{Z}} c_l \varphi\left(x - l\right) \right\|_{L_2} \leq B \cdot \|c\|_{\ell_2}.$$

The proposed formulation covers many signal representation models from the bandlimited one to spline and wavelet-like expansions [5, 77, 133]. The model can be easily extended to multidimensional functions by using tensor-product basis function.

2.2 Approximation methods

We describe in this section how to get the coefficients c_k in (2.2) so that the approximation function, $f_T(x) = Q_T f(x)$ is a reasonable representation of some desired function $f(x)$.

The standard approach consists in prefiltering and sampling thereafter. The method can be described by the inner product integral

$$c_k = \int f(\xi)\tilde{\varphi}\left(\frac{\xi}{T} - k\right) d\frac{\xi}{T} \tag{2.3}$$

where $\tilde{\varphi}(x)$ is called the analysis (or sampling) function.

The linear and shift-invariant approximation operator Q_T is defined as

$$Q_T f(x) = \sum_{k\in\mathbb{Z}} \left[\int f(\xi)\tilde{\varphi}\left(\frac{\xi}{T} - k\right) d\frac{\xi}{T}\right] \varphi\left(\frac{x}{T} - k\right). \tag{2.4}$$

The most critical choice is the synthesis function $\varphi(x)$ because it determines the approximation space V_T. An important property of $\varphi(x)$ is the order of approximation which is related to the rate of decay of the approximation error as a function of the sampling step. We will explain this property and its implications in more detail in the next section.

Once this choice has been made, we need to determine the function $\tilde{\varphi}(x)$ such that the approximation scheme performs appropriately. A widely use criterion is the biorthornomality condition specified by $\langle\tilde{\varphi}(x-k), \varphi(x)\rangle = \delta_k$. A direct consequence of this constraint is that the approximation operator becomes a projector: i.e., $Q_T Q_T = Q_T$, which ensures the exact reproduction of any function $f(x) \in V_T$. Wavelet-like multiresolution approximation operators are examples of such projectors. The standard interpolation scheme corresponds to $\tilde{\varphi}(x) = \delta(x)$. Note that the biorthonormality property is equivalent to the standard interpolation condition $\varphi(k) = \delta_k$, which ensures that the expansion coefficients c_k are the values of the function at the grid points. The least squares approximation which minimizes the sum of the squares of the error, is obtained when $\tilde{\varphi}(x)$ and $\varphi(x)$ are biorthogormals along with the additional constraint $\tilde{\varphi}(x) \in V_1$. The corresponding function $\mathring{\varphi}(x)$ is called the dual of $\varphi(x)$ and is defined in the Fourier domain as

$$\hat{\mathring{\varphi}}(\omega) = \frac{\hat{\varphi}(\omega)}{\hat{a}_\varphi(\omega)} \tag{2.5}$$

where $\hat{a}_\varphi(\omega) = \sum_k |\hat{\varphi}(\omega + 2k\pi)|^2$ is the Fourier transform of the sampled autocorrelation of $\varphi(x)$. This optimal scheme provides an orthonormal projection

which is denoted by P_T instead of Q_T. A special case of this type of approximation operators corresponds to the Shannon's reconstruction formula for a bandlimited signal being $\varphi(x) = \mathring{\varphi}(x) = \text{sinc}(x)$ [110]. We can gain some flexibility by eliminating the constraint $\tilde{\varphi}(x) \in V_1$ while keeping the biorthonormality condition. The price to pay is an increase of the approximation error which can be negligible depending on the angle between the analysis and synthesis space. This suboptimal approximation operator corresponds to an oblique projection.

Finally, we describe a special class of approximation operators: the quasi-interpolators. Very often, in digital signal processing, we make the assumption of ideal sampling in the sense that our discrete signal values represent the true samples $f_k = f(kT)$ of a bandlimited signal $f(x)$. The sampling functions that allow us to stay discrete are a linear combination of Diracs. The corresponding sampling function is

$$\tilde{\varphi}(x) = \sum_{k \in \mathbb{Z}} p_k \delta(x - k)$$

which corresponds to the continuous-time representation of the digital filter p_k. In [21], one can find a clear description of how to calculate the weights p_k for different design criterions. An interpolation operator is the one that satisfies at the same time the quasi-interpolation and the projection conditions.

2.3 Approximation error

***Lth*-order of approximation.** An important concept in approximation theory is the order of approximation which describes the rate of decay of the error as the sampling step goes to zero.

We say that the function $\varphi(x)$ is of order L iff

$$\begin{cases} \hat{\varphi}(0) = 1 \\ \hat{\varphi}^{(l)}(2\pi k) = 0, k \in \mathbb{Z}, k \neq 0 \quad \text{for} \;\; l = 0, ..., L-1 \end{cases}$$

where $\hat{\varphi}(\omega)$ is the Fourier transform of $\varphi(x)$ and $\hat{\varphi}^{(l)}(\omega)$ its lth-derivative with respect to ω. These equations are often referred to as the Strang-Fix conditions of order L [113].

A relevant implication of the order property is the fact that a function $\varphi(x)$ of order L is able to reproduce polynomials of degree 0 to $L-1$. This connection can be established with the help of Poisson's summation formula; for instance, the first-order Strang-Fix condition with $\hat{\varphi}(0) = 1$ is equivalent to the partition of unity: $\sum_k \varphi(x - k) = 1$. In that sense, we can rewrite the Strang-Fix conditions of order L as

$$\begin{cases} 1 = \sum_{k\in\mathbb{Z}} \varphi(x-k) & \text{(Partition of unity)} \\ x^l = \sum_{k\in\mathbb{Z}} \lambda_k^{(l)} \varphi(x-k) & \text{for } \ l = 0, ..., L-1. \end{cases}$$

where the $\lambda_k^{(l)}$ are suitable coefficients.

Lth-**order generating function.** We call an *Lth*-order generating function a continuous function $\varphi(x) \in L_2$ that forms a Riesz basis and has a *Lth*-order of approximation. Note that the generating functions considered here are not necessarily compactly supported [21].

Least-squares approximation error. We are interested in evaluating the quantity

$$\epsilon_f(T) = \|f - Q_T f\|_{L_2} = \left(\int |f(x) - Q_T f(x)|^2 \, dx \right)^{\frac{1}{2}}.$$

The approximation error $\epsilon_f(T)$ can be expressed as a main term plus a perturbation as was demonstrated in [21]. The dominating component can be computed exactly by integration of $|\hat{f}(\omega)|^2$ against the kernel

$$E(\omega) = 1 - \frac{|\hat{\varphi}(\omega)|^2}{\hat{a}_\varphi(\omega)} + \hat{a}_\varphi(\omega) |\hat{\tilde{\varphi}}(\omega) - \hat{\mathring{\varphi}}(\omega)|^2. \tag{2.6}$$

Obviously, the second term of the kernel vanishes for a least-squares approximation (i.e., $\tilde{\varphi}(x) = \mathring{\varphi}(x)$).

The approximation error $\epsilon_f(T)$ is then, given by

$$\forall f \in W_2^r, \epsilon_f(T) = \left[\frac{1}{2\pi} \int E(T\omega) |\hat{f}(\omega)|^2 d\omega \right]^{\frac{1}{2}} + e(f,T) \tag{2.7}$$

where the correction term $e(f,T)$ is bounded as

$$|e(f,T)| \leq K T^r \|f^{(r)}\|_{L_2},$$

where r is the Soboled regularity of $f(x)$ and K is some known constant [22]; $\|f^{(r)}\|_{L_2}$ is the L_2-norm of the *rth* derivative of $f(x)$ which is given by

$$\|f^{(r)}\|_{L_2} = \left[\frac{1}{2\pi} \int \omega^{2r} |\hat{f}(\omega)|^2 d\omega \right]^{\frac{1}{2}}.$$

The correction term $e(f,T)$ can take positive or negative values and becomes very small if the function $f(x)$ is sufficiently smooth.

The result can also be interpreted as a sampling theorem. For the particular case, $\varphi(x) = \mathrm{sinc}(x)$ and $\tilde{\varphi}(x) = \delta(x)$, we recover Shannon's sampling theorem [110].

Averaged approximation error. Suppose that we are approximating a shifted version $f_u(x) = f(x-u)$ of the function $f(x)$. The resulting error is a T-periodic function of the shift increment u, i.e., $\epsilon_{f_{u+T}} = \epsilon_{f_u}$, due to the T-integer shift invariance of the approximation space. As the first term in $\epsilon_f(T)$ (equation (2.7)) is shift-invariant, the influence of the phase factor only appears in its second error component.

We can obtain a delay-independent version of the approximation error by averaging $\|f_u - Q_T f_u\|_{L_2}$ over the period interval T

$$\eta_f^2(T) = \frac{1}{T}\int_0^T \|f_u - Q_T f_u\|_{L_2}^2 du.$$

This quantity was computed exactly in [21] and reduces to the first term in equation (2.7). The averaged approximation error $\eta_f(T)$ corresponds to

$$\eta_f^2(T) = \frac{1}{2\pi}\int E(T\omega)|\hat{f}(\omega)|^2 d\omega. \tag{2.8}$$

This is also the expected approximation error if we consider the initial phase of the sampler random and uniformly distributed.

Asymptotic approximation error. We will use the expression for $\epsilon_f(T)$ to characterize the behavior of the error as the sampling step gets sufficiently small. The main purpose here is to indicate a simple and accurate way of predicting the error so that we can select the interpolation function and the sampling step according to our quality requirements.

We see that as the sampling step tends to zero, we have

$$\epsilon_f(T) = \eta_f(T) + o(T^r).$$

Blu *et al.* in [21] have shown that it is possible to expand the approximation error in a power series of T by taking the Taylor series expansion of the symmetrical kernel $E(\omega)$ around the origin $\omega = 0$. The first nonzero coefficient gives the asymptotic rate of decay of the error $\epsilon_f(T)$ as a function of T. Specifically, if

$\varphi(x)$ and $\tilde{\varphi}(x)$ are quasibiorthonormal of order L: $\varphi(x)$ satisfies the Strang-Fix conditions of order L and the moments of $\tilde{\varphi}(x)$ and $\mathring{\varphi}(x)$ are equal up to the order L (i.e., $\int x^k \tilde{\varphi}(x)dx = \int x^k \mathring{\varphi}(x)dx$ for $k = 0, ..., L-1$). Then, the error will have the characteristic form

$$\epsilon_f(T) = \|f - Q_T f\|_{L_2} = C_L \|f^{(L)}\|_{L_2} T^L$$

as $T \to 0$, where the asymptotic error constant is

$$C_L = \sqrt{\frac{E^{(2L)}(0)}{(2L)!}}. \tag{2.9}$$

This is precisely to what we refer to when we speak of an *Lth*-order approximation scheme. This result specifies a whole class of linear T-integer shift invariant approximation procedures in V_T that provide an $O(T^L)$ error decay rate. Instances of these approximation operators are quasi-interpolators, interpolators and orthonormal and oblique projectors. All these methods with the same rate of decay differ in the value of the approximation constant. For example, if $\varphi(x)$ satisfies the Strang-Fix conditions of order L, $\tilde{\varphi}(x)$ is biorthonormal to $\varphi(x)$ and $\tilde{\varphi}(x)$ satisfies the partition of unity then

$$C_L = \frac{1}{L!} \sqrt{\sum_{k \neq 0} |\hat{\varphi}^{(L)}(2k\pi)|^2}$$

as was stated in [124].

Maximum Order Minimal Support functions. The quality of the approximation of $f(x)$ depends strongly on the order L of the interpolator and no so much on the size of its support N_φ. Nevertheless, N_φ determines the computational cost. The functions that minimize the support N_φ for a given order L are linear combinations of B-spline derivatives which specifies the MOMS (Maximum Order Minimal Support) class of functions

$$\varphi(x) = \sum_{k=0}^{L-1} \gamma_k \frac{d^k}{dx^k} \beta^{L-k}(x) \tag{2.10}$$

where β^{L-k} denotes the B-spline of degree $(L-k-1)$. The reader is referred to the next Chapter 3 for a formal definition of a B-spline. In particular, the

B-splines of degree n are the smoothest functions for a given order of approximation ($L = n + 1$). The asymptotic approximation constant C_L given in (2.3) can be minimized within the MOMS class and the resulting functions are called O-MOMS, where "O" stands for optimal [18]. Among all the $\varphi(x)$'s of support size L, the O-MOMS of order L yields the smallest asymptotic approximation error $\epsilon_f(T)$ independly of the function $f(x)$. They can be determined recursively as indicated in [18, 20]. The cubic O-MOMS, which has approximation order $L = 4$, is given by

$$\varphi_3(x) = \beta^3(x) + \frac{1}{42}\frac{d^2}{dx^2}\beta^3(x).$$

The O-MOMS are continuous but not differentiable for even orders, and even only piecewise continuous for odd orders. So, they are continuous at best. The conclusion is that regularity is not as strong a determinant for the approximation performance as is commonly believed. The value of the asymptotic constant for the cubic B-spline is $C_{4,\beta^3} = (240\sqrt{21})^{-1}$; for $\varphi_3(x)$ we have $C_{4,\varphi_3} = 5040^{-1}$, so that we may expect an 13.22 dB asymptotic improvement when using the cubic O-MOMS instead of the cubic spline[20].

2.4 Summary

In this chapter, we have reviewed some basic concepts in linear approximation theory. All of them will be useful to us later on. More precisely, in digital image processing, there are two major issues related to this topic. On the one hand, signal reconstruction from discrete samples which is commonly achieved by interpolation. On the other hand, projection of a continuously transformed signal onto an appropriate approximation space prior to sampling. In Chapter 7, we will study other aspect of approximation theory, the characterization of the asymptotic approximation error for the more general class of L_p-projections on spline spaces.

Chapter 3

A New Graphical Formalism for Splines

In this chapter, we define a new operator formalism for splines and for multirate signal processing which facilitates the computation of inner products and of convolutions in which B-splines are involved. Thanks to these tools, we are able to simplify many of the derivations for spline-based continuous signal processing. This presentation is meant to be self-contained and may serve as an introduction to the attractive area of splines.

The chapter is organized as follows: First, we present the basic building blocks for our graphical formalism (sampling, scale change and convolution). Second, we derive the graphical representation of some relevant convolutional operators. Third, we introduce the B-splines. These specific basis functions have been chosen due to their remarkable properties. Finally, we apply the graphical formalism we just developed to B-splines, which offers a framework that leads itself naturally to the easy derivation of useful additional relations.

3.1 Basic operators

We consider that all our signals are functions of the continuous space variable x. We modify them using a combination of basic operators which are of three types: sampling, scale change, and convolution. Their definition is as follows:

Sampling. Sampling is the key operation that transforms a continuously defined function $f(x)$ into a discrete signal specified by the samples at the integers $f(x)|_{x=k} = f_k$. It is modeled by a multiplication with a train of Dirac distributions. It is represented as:

$$\iff \quad f(x) \mapsto f_\delta(x) = \textstyle\sum_k f_k \delta(x-k).$$

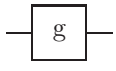
Note that the resulting discrete sequence f_k is mapped in our formalism to the distribution $f_\delta(x)$, which is still defined continuously with respect to x.

Scale change. The magnification-operator is represented as:

$$\iff \quad f(x) \mapsto f\left(\tfrac{x}{a}\right).$$

Thanks to this operator, we can easily model a change of sampling rate by rescaling the signal prior to sampling.

Convolution. A common operator in signal processing is the convolution. In our graphical representation, all the boxes denote impulse responses:



$$\iff \quad f(x) \mapsto g * f(x).$$

The distribution formalism depending on the input and on the filter type the convolution can be implement either a full continuous, a semidiscrete or a full discrete convolution.

3.2 Relevant convolutional operators

We devote this section to the graphical representation of relevant convolutional operators for our formalism. Our tools are the basic operators that were described in the previous section.

Convolution with an arbitrary shift. The linear operator corresponding to an arbitrary shift (i.e., non-integer) is represented as $\boxed{\delta(x-b)}$. Figure 3.1 illustrates the rule for concatenating two shifts and for permuting a shift with a magnification. The proofs are not included because they are trivial.

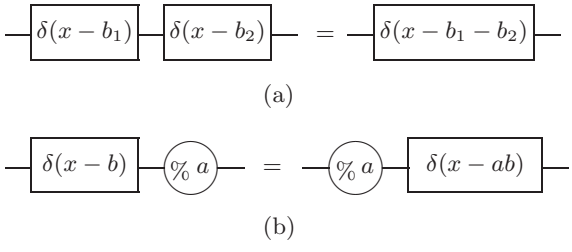


Figure 3.1: Shift by b: (a) Concatenation rule. (b) Permutation with a magnification.

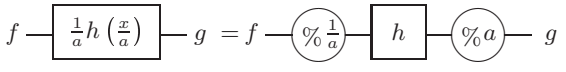


Figure 3.2: Graphical equivalence of the convolution with an expanded signal.

Convolution with an expanded signal. The convolution of a continuous signal $f(x)$ with an expanded version of another continuous signal $h(x)$ deserves to be defined explicitly. The left-hand side of Figure 3.2 corresponds to the convolution

$$g(x) = f(x) * \frac{1}{a}h\left(\frac{x}{a}\right) = \frac{1}{a}\int f(\tau)h\left(\frac{x-\tau}{a}\right)d\tau.$$

If we make the change of variable $u = \frac{\tau}{a}$,

$$g(x) = \int f(a\tau)h\left(\frac{x}{a}-\tau\right)d\tau.$$

we obtain the expression on the right-hand side. This is a key equivalence that facilitates the manipulations when modelling the convolution of a continuous signal with an expanded B-spline.

Convolution with an arbitrary spaced sequence. Closely related to the equivalence defined above is the definition of the mixed convolution of a continuous

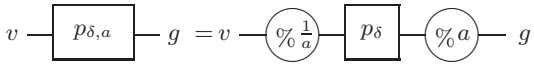


Figure 3.3: Graphical equivalence of the convolution with an arbitrary spaced sequence.

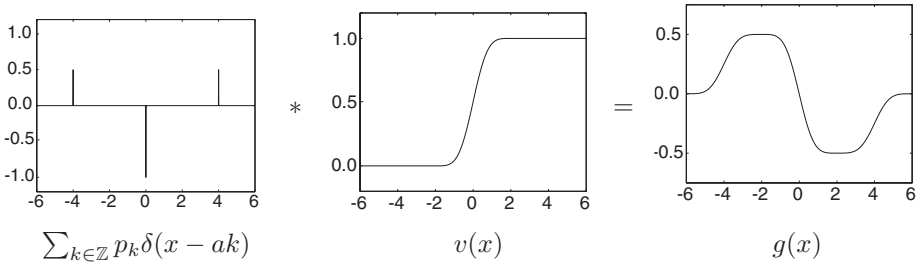


Figure 3.4: Graphical interpretation of the mixed convolution between a discrete sequence p_k and a continuous signal $v(x)$, with $a = 4$.

signal $v(x)$ with a discrete sequence p_k with constant intersample distance a. The reason is that the convolution of a continuous function with an expanded B-spline can be expressed as a mixed convolution, as will be demonstrated later. It is given by

$$g(x) = \underbrace{\left(\sum_{k\in\mathbb{Z}} p_k\delta(x-ak)\right)}_{p_{\delta,a}} * v(x) = \sum_{k\in\mathbb{Z}} p_k v(x-ak), \tag{3.1}$$

where $p_{\delta,a}$ corresponds to a sampling distribution p_δ with a distance a between the samples. It is modelled in our formalism by the rescaling of p_δ as shown in Figure 3.3. The intuitive illustration of this formula is given in Figure 3.4. The mixed convolution is a weighted sum of shifted replicates of the signal $v(x)$ separated by a distance a.

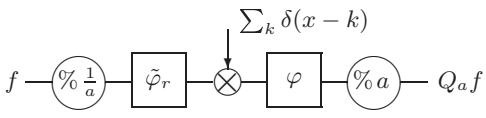


Figure 3.5: Linear shift invariant approximation operator. The symbol $\tilde{\varphi}_r$ denotes the reverse time version of $\tilde{\varphi}$: $\tilde{\varphi}_r(x) = \tilde{\varphi}(-x)$.

Linear shift-invariant approximation operator. As was presented in Chapter 2, the linear shift-invariant approximation operator is defined as

$$Q_a f(x) = \sum_{k\in\mathbb{Z}} \left[\int f(\xi)\tilde{\varphi}\left(\frac{\xi}{a} - k\right) d\frac{\xi}{a} \right] \varphi\left(\frac{x}{a} - k\right) \tag{3.2}$$

Its graphical representation is given in Figure 3.5, uses the equivalence given in Figure 3.2 and illustrates the flexibility of our formalism.

3.2.1 Differential operators

In this section, we assume that all signals $f(x)$ are compactly supported and that all discrete sequences s_k are finite.

Continuous differential operators. The conventional differentiation operator is

$$Df(x) = \frac{df(x)}{dx}.$$

The operator D corresponds to the convolution of $f(x)$ with the derivative of the delta function. We may also write the $(n+1)$-fold derivative operator as

$$D^{n+1} f(x) = \delta^{(n+1)} * f(x).$$

The unique inverse D^{-1} of D is the antiderivative operator

$$D^{-1} f(x) = \int_{-\infty}^{x} f(\tau) d\tau.$$

It is uniquely defined by $DD^{-1}f(x) = D^{-1}Df(x) = f(x)$. The operator D^{-1} also corresponds to the convolution of $f(x)$ with the unit step function. The

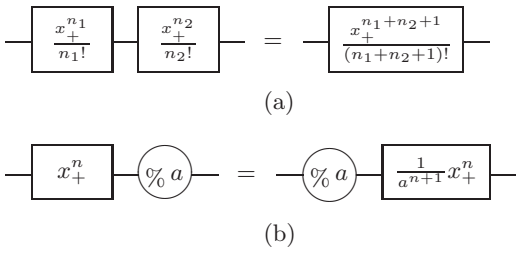


Figure 3.6: One-sided power function: (a) Concatenation rule. (b) Permutation with a magnification.

$(n+1)$-fold convolution of the step function yields the one-sided power function $\frac{x_+^n}{n!}$, where

$$x_+^n = \begin{cases} x^n & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

In particular, the unit step function is x_+^0 with $0_+^0 = \frac{1}{2}$. We may also write the $(n+1)$-fold antiderivative operator as

$$D^{-(n+1)} f(x) = \frac{x_+^n}{n!} * f(x).$$

Note that the concatenation rule $D^{-(n_1+1)} D^{-(n_2+1)} = D^{-(n_1+n_2+2)}$ corresponds to the following composition property of one-sided power functions (see Figure 3.6(a)):

$$\frac{x_+^{n_1}}{n_1!} * \frac{x_+^{n_2}}{n_2!} = \frac{x_+^{n_1+n_2+1}}{(n_1+n_2+1)!}. \tag{3.3}$$

Property. The rule for permuting the magnification operator with the convolution with a one-sided power function is given in Figure 3.6(b). The proof is as follows: We write the expression of the right-hand side in Figure 3.6(b) as

$$\int f\left(\frac{\tau}{a}\right) \frac{1}{a^{n+1}} (x-\tau)_+^n d\tau.$$

We then make the change of variable $u = \frac{\tau}{a}$

$$\int f(u) \frac{1}{a^{n+1}} (x - au)_+^n a \, du = \int f(u) \left(\frac{x}{a} - u \right)_+^n du,$$

which corresponds to the expression on the left-hand side.

In the graphical representation, we will use $\delta^{(n+1)}$ for the differentiation and $\frac{x_+^n}{n!}$ for the antiderivative operators.

Discrete differential operators. We define the backward finite-difference operator as

$$\Delta(x) = \delta(x) - \delta(x - 1).$$

This is also a discrete convolution operator (digital filter), the z-transform of which is

$$\Delta(z) = 1 - z^{-1}.$$

When working with ℓ_1-sequences, we can consider the inverse operator Δ^{-1} defined uniquely as

$$\Delta^{-1}(x) = \sum_{n \geq 0} \delta(x - n).$$

the z-transform of which is

$$\Delta^{-1}(z) = (1 - z^{-1})^{-1}.$$

It can be defined as the running-sum filter

$$(\Delta^{-1} * s)_k = \sum_{n \leq k} s_n,$$

We thus have $\Delta^{-1} * \Delta * s = \Delta * \Delta^{-1} * s = s$.

This operator converges only when it is applied to sequences that are periodic and have a zero average. In the next chapter, we will define a variation of Δ^{-1} that preserves the zero-mean property of the input. We will also model the propagation of boundary conditions through the operator Δ^{-1}. Note that $y_k = \Delta^{-1} * s_k$ can be implemented very efficiently using the recursive equation

$$y_k = y_{k-1} + s_k. \tag{3.4}$$

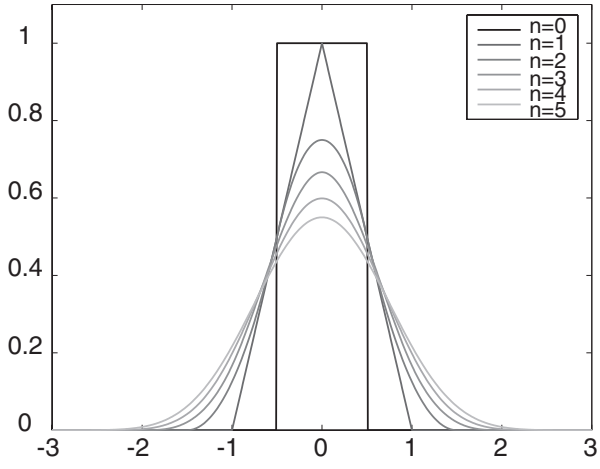


Figure 3.7: The centered B-splines of degree 0 to 5.

Figure 3.8: Schematic representation of a centered B-spline of degree n.

3.3 B-spline definition

The one-sided power function x_+^n, which has a unique singularity of order n at the origin, is a trivial example of a polynomial spline of degree n. While a polynomial spline can always be written as a sum of shifted one-sided power functions, it is more convenient to work with B-splines as basis functions [125]; The noncentered B-splines are obtained through the following finite-difference process:

$$\beta_{nc}^n(x) = \Delta^{n+1} * \frac{x_+^n}{n!}. \tag{3.5}$$

If we shift β_{nc}^n by $\frac{n+1}{2}$, the finite-difference operation is recentered so that the result is a centered B-spline (see Figure 3.7)

$$\beta_c^n(x) = \beta_{nc}^n(x) * \delta\left(x + \frac{n+1}{2}\right). \tag{3.6}$$

Unless mentioned, we will work with centered B-splines that will be noted $\beta^n(x)$. Since a convolution with $\frac{x_+^n}{n!}$ is equivalent to the $(n+1)$-fold antiderivative $D^{-(n+1)}$, we can rewrite the B-spline as

$$\beta^n(x) = \Delta^{n+1} * D^{-(n+1)}\delta\left(x + \frac{n+1}{2}\right). \tag{3.7}$$

Figure 3.8 shows the schematic representation of a B-spline of degree n using our mathematical formalism. Δ^{n+1} is a digital filter, while $\frac{x_+}{n!}$ is a true function of the continuous variable x. The input signal is a Dirac distribution.

The family of B-spline basis functions turns out to be especially attractive for our purpose. Their main advantages are as follows:

1. B-splines are non-negative, symmetric, and compactly supported. They have the shortest support for a given order of approximation $L = n + 1$, which means that the computational complexity is minimized [20].

2. B-splines satisfy the partition of unity and have excellent approximation properties [22]. A B-spline of degree n is part of the MOMS family of order L.

3. Splines are smooth and well-behaved (piecewise polynomials).

4. The simple analytic form of splines facilitates their manipulation [125].

5. They satisfy a two-scale relation which makes them appropriate for multiscale processing [124].

6. The family of polynomial splines also provide design flexibility. By increasing the degree, we can progressively switch from simple polynomial representations (piecewise constant $(n = 0)$ or linear $(n = 1)$) to the bandlimited model $(n \to \infty)$[7].

3.4 B-spline relations

As we have just seen, B-spline basis functions possess a number of nice properties that make them well-suited for the computation of projection-based image

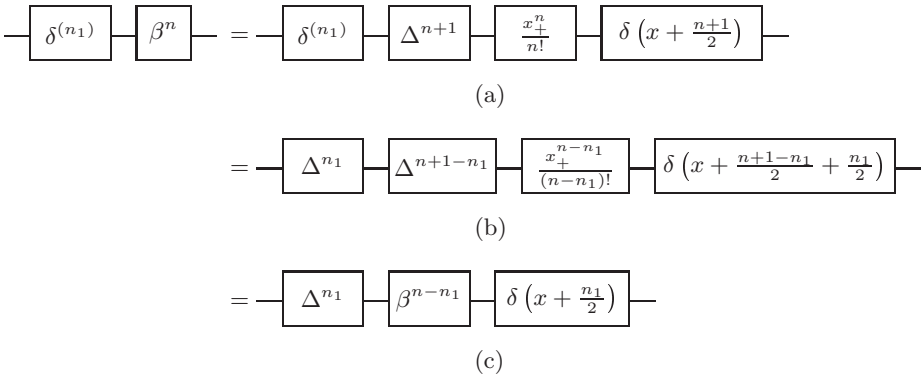


Figure 3.9: Graphical derivation of the n_1-fold derivative of a B-spline. (a) Substitution of β^n by its explicit expression given in Figure 3.8. (b) Application of the commutativity of "$*$" and the identity $\Delta^{n_1} * \Delta^{-n_1} = \mathbb{I}$. (c) Scheme obtained using the definition of the B-spline.

approximations. As was explained in the introduction, the main difficulty of the algorithms is that they involve the computation of inner products (or convolutions) with B-splines of different widths. In this section, we propose an exact solution for these type of computations applicable to B-splines of any order and arbitrary size.

3.4.1 Differentiation

As an example of the fourth property of Section 3.3, we derive the two most relevant B-spline differential relations:

- n_1-th derivative for $n_1 \leq n$:

$$D^{n_1}\beta^n(x) \quad = \quad \Delta^{n_1} * \beta^{n-n_1}\left(x+\frac{n_1}{2}\right). \tag{3.8}$$

Proof: In Figure 3.9(a), β^n is substituted by its explicit expression given in Figure 3.8. Using $\Delta^{n_1} * \Delta^{-n_1} = \mathbb{I}$ and the commutativity of "$*$",

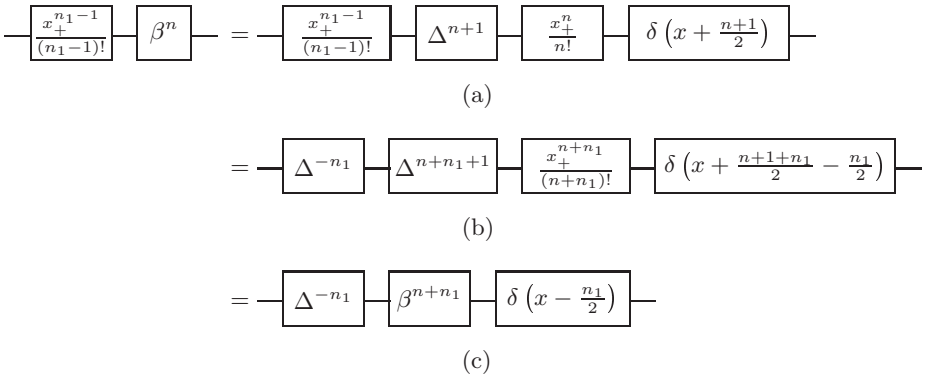


Figure 3.10: Graphical derivation of the n_1-fold integral of a B-spline. (a) Substitution of β^n by its explicit expression given in Figure 3.8. (b) Application of the commutativity of "$*$" and the identity $\Delta^{n_1} * \Delta^{-(n_1)} = \mathbb{I}$. (c) The proof is finished by using the explicit time domain expression for B-splines.

we obtain Figure 3.9(b). We complete the proof by identifying the explicit time domain expression of β^{n-n_1}.

■

- n_1-fold integral for $n_1 \leq n$:

$$D^{-n_1}\beta^n(x) \quad = \quad \Delta^{-n_1} * \beta^{n+n_1}\left(x - \frac{n_1}{2}\right). \tag{3.9}$$

Proof: The proof is derived graphically in Figure 3.10. It follows the same steps as the one for the n_1-fold derivative.

■

3.4.2 Expanding B-splines

As we commented in the introduction, the convolutive equivalence of $\beta_a(x)$ is a essential tool in our formalism. The resulting explicit time-domain expression for an expanded B-spline as derived in Figure 3.11 is

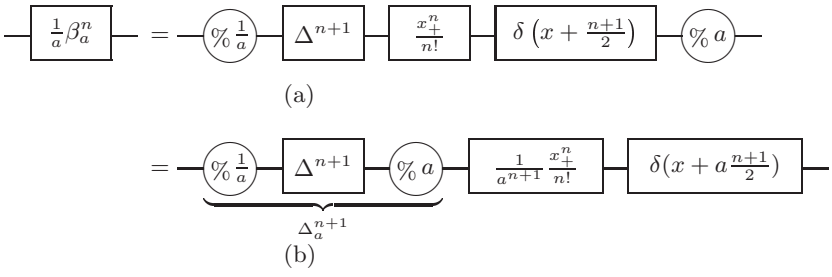


Figure 3.11: Diagram that shows how to derive the analytic expression for the B-spline expanded by a factor a. (a) Substitution of β^n by its time-domain explicit expression (see Figure 3.8). (b) Application of the permutation rules for the shift and the one-sided power function (see Figures 3.1(b) and 3.6(b)).

$$\frac{1}{a}\beta_a^n(x) = \frac{1}{a}\beta^n\left(\frac{x}{a}\right) = \Delta_a^{n+1} * \frac{1}{a^{n+1}}\frac{x_+^n}{n!} * \delta\left(x + a\frac{n+1}{2}\right), \tag{3.10}$$

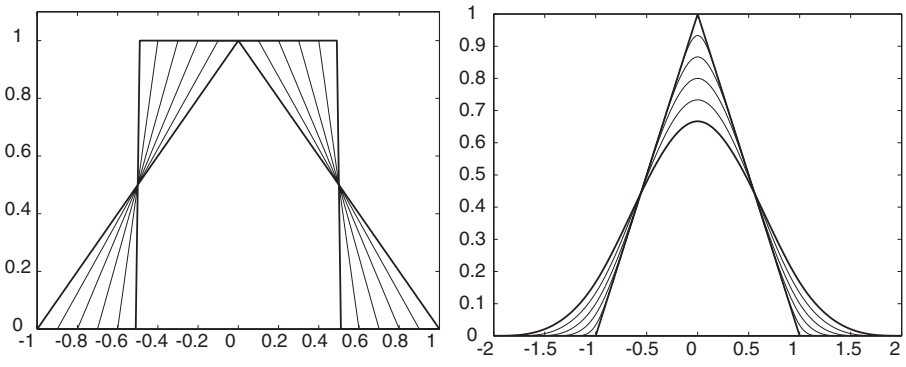
where Δ_a is the rescaled finite-difference operator

$$\Delta_a^{n+1} = \sum_{k=0}^{n+1} \underbrace{\binom{n+1}{k}(-1)^k}_{q(k)} \delta(x - ak). \tag{3.11}$$

Note that $\lim_{a\to 0} \frac{1}{a}\beta_a^n(x) = \delta(x)$.

3.4.3 Spline bikernels

We define the spline bikernels as the convolution of two B-splines of variable widths and degrees (see Figure 3.12). The explicit time domain expression of a spline bikernel is as follows:



(a) (b)

Figure 3.12: Spline bikernels computed by the convolution of the two B-splines $\frac{1}{a}\beta^{n,n}_{1,a} = \beta^n_1 * \beta^n_a$ with $a \in [0,1]$. (a) $n = 0$; (b) $n = 1$. The kernels generate a smooth transition between B-splines of degree n and degree $2n+1$. For $a = 0$ and $a = 1$, the kernels are B-splines because $\lim_{a\to 0} \frac{1}{a}\beta^n_1 * \beta^n_a = \beta^n_1$ and $\beta^n_1 * \beta^n_{a=1} = \beta^{2n+1}_1$.

$$\begin{aligned}\frac{1}{a_1 a_2}\beta^{n_1,n_2}_{a_1,a_2}(x) &= \frac{1}{a_1 a_2}\beta^{n_1}_{a_1}(x) * \beta^{n_2}_{a_2}(x) \\ &= \Delta^{n_1+1,n_2+1}_{a_1,a_2} * \frac{1}{a_1^{n_1+1}a_2^{n_2+1}}\frac{x_+^{n_1+n_2+1}}{(n_1+n_2+1)!} \\ &\quad * \delta\left(x + a_1\frac{n_1+1}{2} + a_2\frac{n_2+1}{2}\right),\end{aligned} \tag{3.12}$$

where we denote the convolution of two finite-difference operators of different widths as

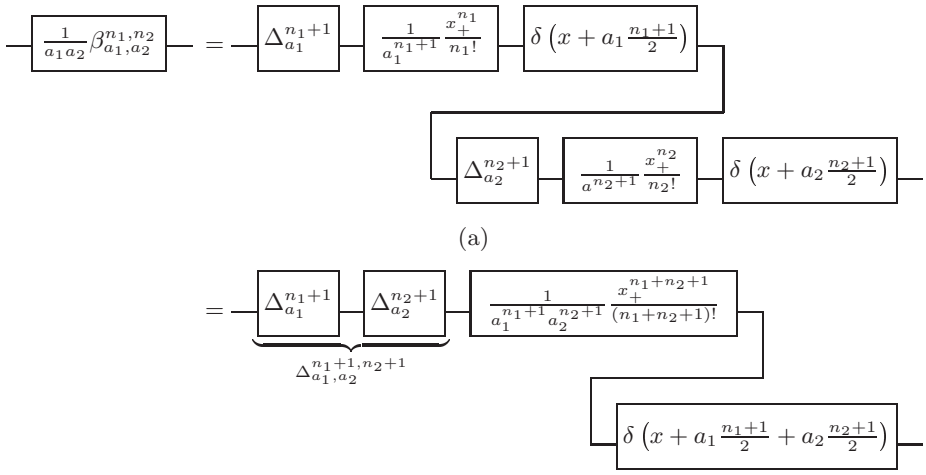


Figure 3.13: Time domain explicit expression for the spline bikernel $\frac{1}{a_1 a_2}\beta^{n_1,n_2}_{a_1,a_2}$. (a) Substitution of $\frac{1}{a_1}\beta^{n_1}_{a_1}$ and $\frac{1}{a_2}\beta^{n_2}_{a_2}$ by its expression given in Figure 3.11. (b) Application of the concatenation rule of one-sided power functions and of shifts (see Figures 3.1(b) and 3.6(b)).

$$\begin{aligned}\Delta^{n_1+1,n_2+1}_{a_1,a_2} &= \Delta^{n_1+1}_{a_1} * \Delta^{n_2+1}_{a_2} \\ &= \sum_{k=0}^{n_2+1}\sum_{l=0}^{n_1+1} q_2(k)q_1(k-l)\delta(x - a_1 l - a_2 k) \qquad (3.13)\end{aligned}$$

The knots positions of the non-uniform spline that constructs the spline bikernel $\beta^{n_1,n_2}_{a_1,a_2}$ are $\{a_2 k + a_1 l\}$ for $k = 0, ..., n_2 + 1$ and $l = 0, ..., n_1 + 1$. Note that if $a_1 = a_2 = a$, then $\Delta^{n_1+1}_a * \Delta^{n_2+1}_a = \Delta^{n_1+n_2+2}_a$.

The graphical derivation on Figure 3.13 is based on the convolutive equivalent of the B-spline expanded by an arbitrary factor a (see Figure 3.11). The final expression is simply obtained applying the concatenation rule of shifts and one-sided power functions (see Figures 3.6(b) and 3.1(b)).

We have used these kernels in two applications: image resizing and computer tomographic reconstruction [59].

3.4.4 B-spline inner products

In this section, our concern is to derive an explicit formula for computing B-spline inner products and convolutions, as they are the basis for our projection based image approximations. As a B-spline inner product is equivalent to a

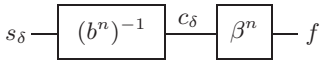


Figure 3.14: The spline f interpolates the input samples s_k.

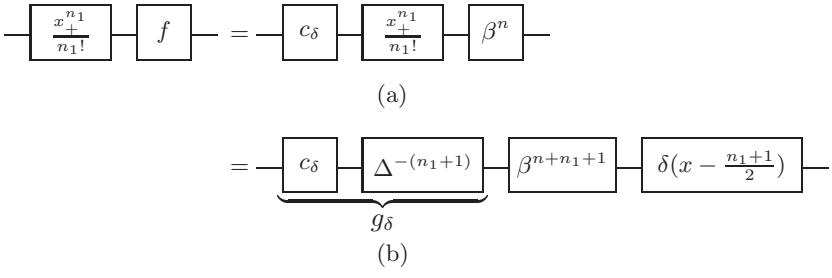


Figure 3.15: Schematic representation of the (n_1+1)-fold integral of a signal expressed in a B-spline basis. (a) Substitution of $f = c_\delta * \beta^n$. (b) The final expression is derived using the (n_1+1)-integral relation for a B-spline (see Figure 3.10).

sampled convolution

$$\frac{1}{a}\left\langle f(x), \beta^{n_1}\left(\frac{x}{a}-k\right)\right\rangle = \frac{1}{a}\left(f * \beta_a^{n_1}\right)(ka),$$

we will derive an exact formula to compute $(f * \beta_a^{n_1})(x)$ with $f(x)$ being a spline. By substitution of the definition of an expanded B-spline (see Figure 3.11), we get

$$\frac{1}{a}(f * \beta_a^{n_1})(x) = \frac{1}{a^{n_1+1}}\Delta_a^{n_1+1} * D^{-(n_1+1)} f\left(x + a\frac{n_1+1}{2}\right). \tag{3.14}$$

Thus, we can compute the convolution in a rather simple way by applying finite differences to the (n_1+1)-fold antiderivative of a shifted version of the function $f(x)$. What makes an exact computation possible and analytically tractable is the fact that the (n_1+1)-fold integral of a spline is a spline with a corresponding increase of the degree. See Figure 3.16(a) for the graphical derivation of equation (3.14).

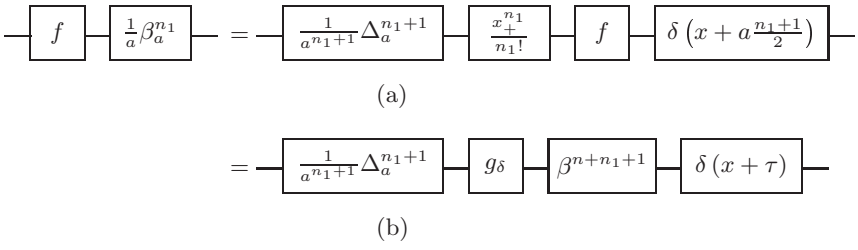


Figure 3.16: B-spline convolution. (a) Substitution of β_a^n by its expression as given in Figure 3.11. (b) Substitution of $\frac{x_+^{n_1}}{n_1!} * f$ by its formula given in Figure 3.15 with $\tau = (a-1)\left(\frac{n_1+1}{2}\right)$.

An alternative interpretation of this equation as a mixed convolution is obtained by replacing $\Delta_a^{n_1+1}$ by its definition

$$\frac{1}{a}(f * \beta_a^{n_1})(x) = \left(\sum_{k=0}^{n_1+1} p_k \delta(x-ak)\right) * v(x), \tag{3.15}$$

where the coefficients p_k are simply a scaled version of the binomial coefficients q_k, $p_k = \frac{1}{a^{n_1+1}} q_k$, and where the continuous function $v(x)$ is the (n_1+1)-fold antiderivative of a shifted version of the function $f(x)$

$$v(x) = D^{-(n_1+1)} f\left(x + a\frac{n_1+1}{2}\right). \tag{3.16}$$

Now, we will derive an expression for the case where $f(x)$ is a spline that interpolates the discrete input samples $s(k)$. Consequently, we have $f(x) = c_\delta * \beta^n(x)$ with $c_\delta = s_\delta * (b_\delta^n)^{-1}$, where $(b^n)^{-1}$ is the inverse filter of the B-spline interpolation filter $b^n = \beta^n(x)|_{x=k}$, as show in [125] (see Figure 3.14). Then,

$$D^{-(n_1+1)} f(x) = g_\delta * \beta^{n+n_1+1} * \delta\left(x - \frac{n_1+1}{2}\right) \tag{3.17}$$

where g_δ is obtained by iterative summation of the interpolation coefficients

$$g_\delta = \Delta^{-(n_1+1)} * c_\delta. \tag{3.18}$$

as shown in Figure 3.15. Finally, the B-spline convolution can be rewritten by substitution of equation (3.17) onto equation (3.16) as the mixed convolution given in Figure 3.16(b) where $v(x)$ is a spline of degree $(n + n_1 + 1)$

$$v(x) = g_\delta * \beta^{n+n_1+1} * \delta(x + \tau), \tag{3.19}$$

and where τ is the shift

$$\tau = (a - 1)\left(\frac{n_1 + 1}{2}\right).$$

3.5 Summary

In this chapter, we have constructed a mathematical formalism that allows to compute exact inner products (or convolutions) involving B-splines of different widths using finite differences.

In the next chapters, we will use this catalog of mathematical operations to derive discrete algorithms that solve in an innovative and exact fashion common problems in image and signal processing. These include resizing by an arbitrary scale, performing continuous wavelet transforms, and non-uniform to uniform resampling.

One of the advantages of the method is that the corresponding algorithms can be implemented using digital filters and matrix multiplications. In many cases, the values of the splines or of the polynomial functions involved in the matrix multiplications are signal-independent so, by computing them once and by storing the result in a look-up table, the computational complexity can be greatly reduced.

Chapter 4

Least-Squares Image Resizing

Image resizing (magnification or reduction) is a common operation in image processing [94]. It is used whenever one wants to change the image resolution. For example, it is required on a routine basis in digital photography, multi-media and electronic publishing [72, 105], for adapting the pixel size to the resolution of an output device (printer or monitor) [48, 56], and for generating preview images, or posting digital pictures on the Web.

Another important area of applications is medical imaging; typical instances are:

- Reslicing for resolution normalization. This is to compensate for the fact that 3D volumetric data (CT, SPECT or MRI) are often acquired in a non-isotropic fashion—the within-slice resolution is typically finer than the across-slice resolution [13].
- Image zooming. It is often used to focus on details for diagnostic purposes.
- Image pyramids for multi-scale processing. Many iterative image processing algorithms can be applied in a coarse-to-fine fashion. Working with smaller images reduces the computation time and also tends to improve robustness [71].

Many linear resizing techniques are available even though they have some limitations. The standard ones rely on interpolation [119]. The simplest methods are nearest-neighbor and bilinear interpolation, which correspond to fitting the image with a spline of degree 0 and 1, respectively. The piecewise constant model generates noticeable blocking artifacts, while the (bi-)linear one

tends to lose details through image blurring. Better interpolation performance is achieved by switching to higher order models [119]; typical examples are Keys' short kernel convolution [68], or higher order spline interpolation which offers a better cost-performance ratio [19, 80, 125]. While interpolation works well for image magnification, it is not entirely suitable for image reduction because of potential aliasing problems. The standard remedy is to apply some kind of lowpass prefiltering prior to resampling. Although a complete suppression of aliasing is possible through the application of Shannon's ideal filter, this is not a widely used technique—it is computationally expensive and tends to introduce ringing artifacts (Gibbs oscillations).

The principal limitation of interpolation approaches is that they are not designed to minimize information loss. It therefore makes good sense to investigate the possibility of obtaining the best solution in the least-squares sense [131]. Indeed, the signal-to-noise ratio (SNR) is a standard figure of merit used in image processing.

The least-squares solution is achieved by modifying the interpolation approach so that the resampling step gets replaced by the evaluation of inner products with the translates of a suitable analysis function $\tilde{\varphi}$. This computation is equivalent to applying a continuously-defined prefilter (anti-alias) to the interpolated function prior to resampling—the prefilter is not necessarily ideal but is chosen to be biorthogonal to the underlying interpolation kernel. While the basic principle of this projection method was introduced in [131], an exact least-squares implementation was only demonstrated for splines of degree 0 and 1. As we already know, the practical limitation is the difficulty to perform an exact numerical implementation of the optimal prefilter for higher order splines. Lee *et al.* developed a higher order spline resizing algorithm by replacing the orthogonal projection of [131] by an oblique one [73]. They simplified the procedure by replacing the optimal prefilter by a box function analysis (B-spline of degree 0) and made it more efficient by pre-computing the anti-derivative of the function to be approximated.

Here, we present a generalization of this method that allows us to compute both oblique and orthogonal projections (least-squares approximations) for splines of any degree n. What makes the approach feasible in this more general setting is the new finite difference method presented in Chapter 3: it allows an *exact* computation of the required inner products for analysis functions that are B-splines of any degree n. The method derived in Section 4.1 and 4.2 works for both reduction and magnification of images with an arbitrary scaling factor and for any translation value. In Section 4.3, we generalize the approach to a whole class of piecewise polynomial functions (all linear combinations of

B-splines), including some with optimal approximation properties. In Section 4.4, we compare the approximation errors of the least-squares, interpolation and oblique projection method. The quality of the approximation, which depends on the type of algorithm and on the degree of the B-spline, is also characterized in this section.

4.1 Philosophy of the approach

The approach follows the philosophy that was presented in Chapter 2.

As the image is represented using separable basis functions, the resizing problem can be solved optimally in a separable fashion. Consequently, the complexity is reduced from 2-D to 1-D.

The schematic continuous-space domain representation of the whole algorithm using the operator formalism presented in Chapter 3, is given in Figure. 4.2. As we know already, all boxes denote convolutions; g_δ, q_δ and h_δ are digital filters, while φ and φ_1 are analog filters with impulse response $\varphi(x)$ and $\varphi_1(x)$, respectively. The affine transformation $s\left(\frac{x}{a}+b\right)$ is represented via the combination of a shift (convolution with $\delta(x+b)$), and of a resizing $s(x) \rightarrow s\left(\frac{x}{a}\right)$ represented as $-\!\!\left(\%a\right)\!\!-$.

We now describe the four main steps of the method.

4.1.1 Interpolation

The first step is to take the discrete input data s_k and to construct a continuous interpolating model $s(x) = \sum_k c_k \varphi(x-k)$, where the $\varphi(x-k)$'s are some specified basis functions. For this purpose, we take the samples s_k and convolve them with an appropriate prefilter g_k to get the coefficients $c_k = g_k * s_k$. The continuous-time function is obtained by convolving $c_\delta(x)$ with $\varphi(x)$. The prefilter $G(z) = \sum_k g_k z^{-k} = \frac{1}{\sum_k \varphi_k z^{-k}}$ is the convolution inverse of the sequence $\varphi(k)$. This particular choice ensures that the interpolation requirement is satisfied; i.e., $s(x)|_{x=k} = s_k$.

4.1.2 Affine transformation (conceptual step)

We apply the affine transformation (scaling and shifting) to the function $s(x)$.

$$f(x) = s\left(\frac{x}{a}+b\right)$$

The image is enlarged if $a > 1$ and shrunk if $a < 1$.

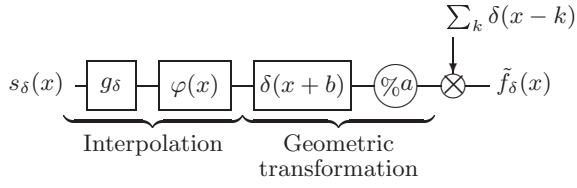


Figure 4.1: General scheme for the standard interpolation approach

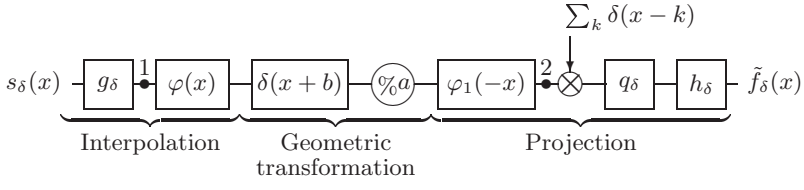


Figure 4.2: General scheme for the proposed projection method

4.1.3 Projection-based signal approximation

In the standard interpolation approach, the image gets resized by resampling $f(x)$ at the integers [60], as illustrated in Figure. 4.1. Here, we will consider an alternative approach in terms of projection operators. Specifically, we find the best approximation $\tilde{f}(x) \in V_{\varphi_2}$ of $f(x)$ in some space $V_{\varphi_2} = \text{span}_k\{\varphi_2(x-k)\}$ such that the L_2-approximation error $\epsilon_a(f) = \|f - \tilde{f}\|_{L_2}$ is minimized. From the Projection Theorem, we know that the least-squares solution to this problem is the orthogonal projection of $f(x)$ onto V_{φ_2} (see [5] or Section 2.2).

$$\tilde{f}(x) = Pf(x) = \sum_k c_2(k)\varphi_2(x-k) = c_{2,\delta} * \varphi_2(x),$$

with $c_2(k) = \langle f, \mathring{\varphi}_2(x-k)\rangle$, where $\mathring{\varphi}_2(x)$ is the dual of the analysis function $\varphi_2(x)$; in other words, $\mathring{\varphi}_2$ satisfies $\mathring{\varphi}_2 \in V_{\varphi_2}$ and $\langle \mathring{\varphi}_2(x), \varphi_2(x-k)\rangle = \delta_k$.

Rather than computing the inner product $\langle f, \mathring{\varphi}_2(x-k)\rangle$, we consider a slightly more general and also more flexible approach via the block diagram in Figure. 4.2. It corresponds to an *oblique* projection onto V_{φ_2} It uses an auxiliary analysis function $\varphi_1(x)$ which is essentially arbitrary.

First, we compute the inner products

$$c_1(k) = \langle f(x), \varphi_1(x-k) \rangle, \tag{4.1}$$

which is equivalent to prefiltering $f(x)$ with $\varphi_1(-x)$ and sampling thereafter.

The cross-correlation sequence of $\varphi_1(x)$ and $\varphi_2(x)$ is given by $a_{12}(k) = \langle \varphi_1(x), \varphi_2(x-k) \rangle$. If $a_{12}(k) \neq \delta_k$, the projection of $f(x)$ onto V_{φ_2} perpendicular to V_{φ_1} requires an additional digital filtering correction q to satisfy the biorthogonality condition [126]. Thus, $c_2(k) = c_1(k) * q(k)$. The appropriate correction filter q is the convolution inverse of a_{12}: $q = a_{12}^{-1} \leftrightarrow \frac{1}{\sum_k a_{12}(k) z^{-k}}$. This is equivalent to using the analysis function $\tilde{\varphi}_2(x) = \sum_k q_k * \varphi_1(x-k) = q_\delta * \varphi_1(x)$, where $\varphi_2(x)$ and $\tilde{\varphi}_2(x)$ are biorthonormal. If $\varphi_1(x) \in V_{\varphi_2}$, then we get the orthogonal projection; otherwise, we have an oblique projection [126].

When computing the orthogonal projection, we obtain a resized image with minimum loss of information in the least-squares sense. If instead, we choose an oblique projection, the approximation is only slightly suboptimal, depending on the angle between $V_{\varphi_1} = \text{span}_k\{\varphi_1(x-k)\}$ and $V_{\varphi_2} = \text{span}_k\{\varphi_2(x-k)\}$ [126]. Moreover, the rate of convergence depends on the approximation order properties of the synthesis function alone; the analysis function has essentially no influence on the asymptotic approximation error [124].

4.1.4 Resampling of the projection at the integers

Finally, we have to resample the projection at the integers ($\tilde{f}(l) = \tilde{f}(x)|_{x=l}$) to get the output of the system $\tilde{f}_\delta(x) = \sum_l \tilde{f}(l)\delta(x-l)$. This is achieved by postfiltering with $h_k = \varphi_2(k)$, the sampled version of the synthesis function.

If we compare the block diagram in Figures. 4.1 and 4.2, we see that the standard interpolation approach corresponds to the simplified situation where $\varphi_1(x) = \delta(x)$ and $q_k * h_k = \delta_k$. The main difficulty with our new approach is the computation of the inner products $\langle f(x), \varphi_1(x-k) \rangle$ involving continuously-defined functions that are specified on different grids.

4.2 Spline resizing algorithm

4.2.1 Derivation of the algorithm

As justified in Chapter 3, we choose our basis functions to be B-splines. In that case, our algorithm has the following parameters: $\varphi(x) = \beta^n(x)$, $\varphi_1(x) = \beta^{n_1}(x)$

and $\varphi_2(x) = \beta^{n_2}(x)$. This implies that $a_{12}(k) = q^{-1}(k) = \beta^{n_1+n_2+1}(x)|_{x=k}$ and $\phi_2(x) = q_\delta * \beta^{n_2}(x)$.

In the sequel, we will derive our final form of the resizing algorithm graphically by using the exchange rules for the one-sided power functions and for the shift given in Figures. 3.1(b) and 3.6(b), together with the convolution rule for one-sided power functions given in Figure 3.6(a).

We now proceed by successive modifications of the block diagram in Figure 4.2. We have extracted the operators between the marks 1 and 2 for simplicity. The final result is shown in Figure 4.3-e.

In Figure 4.3-a, $\varphi(x)$ and $\varphi_1(-x)$ are substituted by their explicit expression using (3.6). Using the rules in Figure 3.1(b) and 3.6(b), the boxes are reorganized in such a way that all one-sided power functions and shifts are moved to the left side of the scale change $-\!\!\left(\%_0 a\right)\!\!-$ as shown in Figure 4.3-b. In this way, the support of the resampling kernel does not depend on a.

The rule for the convolution of one-sided power functions (3.3) is applied to get Figure 4.3-c. Using $\Delta^{n_1+1} * \Delta^{-(n_1+1)} = I$ and $\Delta^{n+1} * \Delta^{n_1+1} = \Delta^{n+n_1+2}$ we obtain Figure 4.3-d. The explicit time domain expression for B-splines is the key to get Figure 4.3-e with the final expression for the spline kernel being $\phi(x) = a^{n_1+1}\beta^{n+n_1+1}(x + \tau_2)$ with $\tau_2 = \frac{n_1+1}{2}\left(\frac{1}{a} - 1\right) + b$. Note that we are allowed to push the sampling step towards the resizing box because the filters located at positions 4 and 5 are all digital.

4.2.2 Practical implementation

We now briefly summarize the main steps in the implementation of the method:

1. Digital prefiltering with the (symmetric) exponential filter $g = (b^n)^{-1}$ to get c_k (interpolation coefficients) from s_k (input samples). The filter is implemented recursively using a cascade of simple causal and anti-causal operators as described in [128].

2. $(n_1 + 1)$-running sums corresponding to the operator $\Delta^{-(n_1+1)}$; these are computed recursively as well by iterating (3.4). The next section describes an implementation that handles the boundary conditions correctly.

3. Geometric transformation and resampling using a spline interpolation model of degree $(n + n_1 + 1)$ (basis function $\phi(x)$).

4. $(n_1 + 1)$-centered finite differences, corresponding to the operator Δ^{n_1+1}.

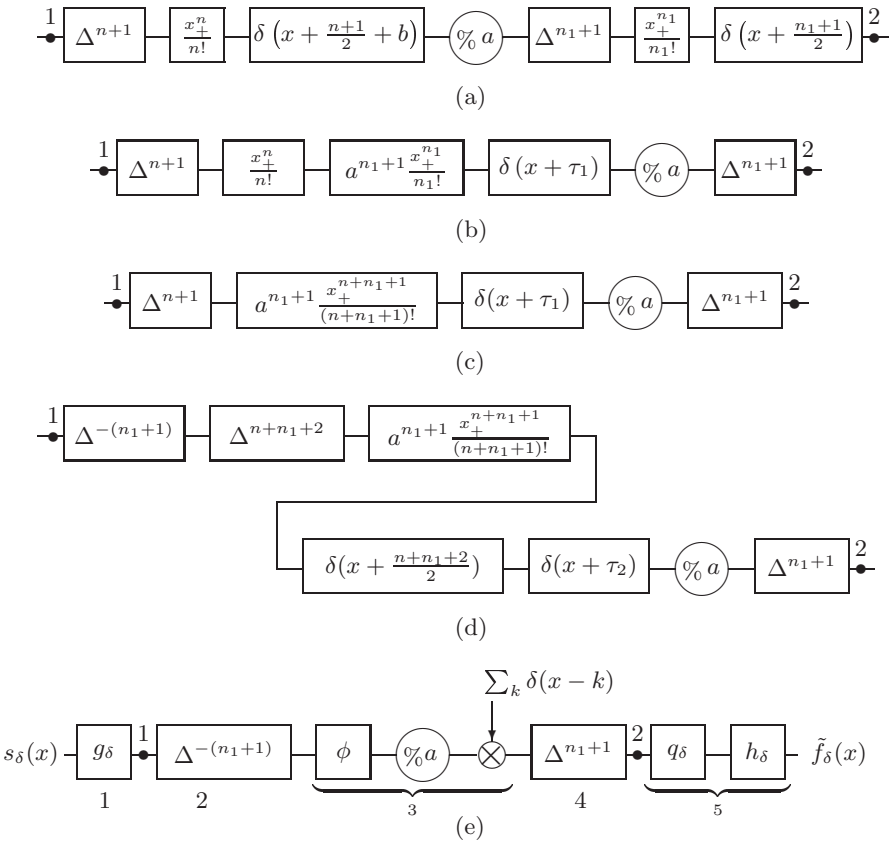


Figure 4.3: Diagram that shows the full process to get our algorithm. (a) Substitution of $\beta^n(x)$ and $\beta^{n_1}(-x)$ by their explicit time expression. (b) Scheme obtained using the exchange rule for $\frac{x_+^n}{n!}$ and $\delta(x+b)$, with $\tau_1 = \frac{n+1}{2} + \frac{n_1+1}{2a} + b$. (c) Application of convolution rule of $\frac{x_+^n}{n!}$ (see Figure 3.6(b)). (d) Application of $\Delta^{n_1+1} * \Delta^{-(n_1+1)} = I$, $\Delta^{n_1+1} * \Delta^{n+1} = \Delta^{n+n_1+2}$ and $\tau_2 = \frac{n_1+1}{2}\left(\frac{1}{a}-1\right) + b$. (e) Equivalent form of Figure. 1 with $\phi(x) = a^{n_1+1}\beta^{n+n_1+1}(x+\tau_2)$, $g = (b^n)^{-1}$, $q = a^{-1} = (b^{n_1+n_2+1})^{-1}$ and $h = b^{n_2}$. The numbers below the diagram indicate the main steps in the implementation.

5. Digital postfiltering with the sampled synthesis function $\phi_2(x)|_{x=k} = q_k * h_k$, where q is an IIR filter implemented using the recursive routines developed in [128].

4.2.3 Computational issues

We can easily trade computational speed against image quality. The most important choice is the underlying signal model (the spline degree n) which determines the approximation properties of the solution. The second parameter, n_1, can be selected to obtain the optimal least-squares solution ($n_1 = n$), or a slightly suboptimal one which corresponds to oblique projection ($-1 \leq n_1 < n$). For the limiting case $n_1 = -1$, we recover the traditional interpolation approach provided that we define the B-spline of degree -1 as the Dirac delta distribution (ideal sampler). The larger $n_1 \leq n$, the better the quality but at the expensive of more computations.

The expensive part of the algorithm is the resampling with the kernel ϕ (step 3 in Section IV.B), which is equivalent to a spline interpolation of degree $(n + n_1 + 1)$. The cost of the rest of the procedure is negligible in comparison: it involves digital filtering only—either short kernel FIR or fast recursive IIR. Thus, we can consider that the total cost per computed output point is proportional to $(n + n_1 + 1)$ times the number of operations required to evaluate ϕ (B-spline of degree $(n + n_1 + 1)$).

One practical limitation of the present approach is the potential propagation of roundoff errors during the multiple integration process. This requires working with high precision arithmetic. Our implementation uses the double type in C and can handle values up to $n_1 = 4$ with typical image of size $512 * 512$.

4.2.4 Boundary conditions and discrete differential operators

The only remaining issue of the chapter is the extension to periodic signals and the study of how to handle consistently the boundary conditions when implementing the algorithms based in our finite differences method. The difficulty comes from the running sum filters $\Delta^{-(n+1)}$ which, in principle, are neither symmetric nor anti-symmetric.

Signal extensions

We will consider periodization and the two types of boundary conditions shown in Figure 4.4.

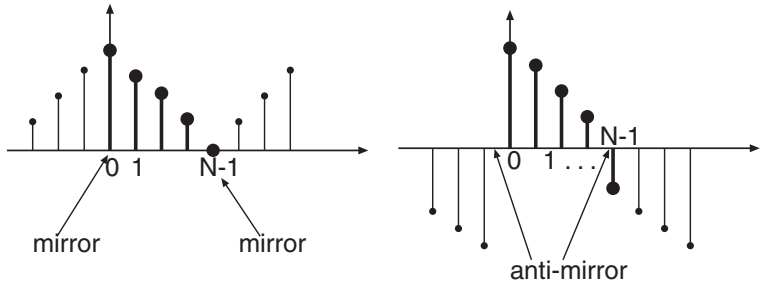


Figure 4.4: Signal extended using: (left) symmetric boundary conditions; (right) anti-symmetric boundary conditions.

- Periodization

  The implementation of the digital filter $\Delta^{-(n+1)}$ is undefined for periodic signals as it attempts to compute an infinite sum of periodic data. A way to overcome this difficulty is to restrict ourselves to periodic signals that have a *zero mean*, i.e., such that $\sum_k s(x-k) = 0$, and to express the P-periodic signal $s(x)$ as $\lim_{m\to\infty} s_{[-mP,(m+1)P]}(x)$ where $s_{[-mP,(m+1)P]}(x)$ is the restriction of $s(x)$ to the support $[-mP,(m+1)P]$. The compactly supported signal $s_{[-mP,(m+1)P]}(x)$ can also be expressed by a convolution

$$s_{[-mP,(m+1)P]}(x) = s_{[0,P]}(x) * \underbrace{\sum_{|k|\leq m} \delta(x-kP)}_{\Pi_m(x)}.$$

  Then, since Δ^{-1} commutes with the periodization operator $\Pi_m(x)$ and, because $s_{[0,P]}(x)$ has zero mean, $\Delta^{-1} * s_{[0,P]}$ is also finitely supported within $[0,P]$ (more precisely: within $[0,P-1]$). This implies that Δ^{-1} keeps its meaning as m tends to infinity. If $s(x) = \sum_k s_k \delta(x-k)$, where $s_k = s_{k+P}$ (periodicity) and $\sum_{k=0}^{P-1} s_k = 0$ (zero-mean requirement), then we have

$$(\Delta^{-1} * s)_k = \sum_{l=0}^{k \bmod P} s_l.$$

  Since we have to apply the operator Δ^{-1} repeatedly, a process that does

not preserve the zero-mean property, we will indicate in the next subsection how to enforce this property on any periodic input.

- Symmetry

  To minimize boundary artifacts, we extend our signal $\{s_k\}_{k=0,\dots,N-1}$ using symmetric mirror boundary conditions defined as $s_{-k} = s_k$, and $s_{N-1-k} = s_{N-1+k}$, for $k = 0, 1, \dots, N-1$. This process is repeated on the newly extended signal, $\{s_k\}_{k=-N+1,\dots,2N-3}$ and so further. As can readily be verified, this is equivalent to requiring that $s(-x) = s(x)$ and that $s(x)$ is $(2N-2)$-periodic. In other words, it is sufficient to specify what happens around the origin; the symmetry on the other end is propagated automatically through the periodization process.

- Anti-symmetry

  Another complementary technique interesting to us because it is satisfied by signals that appear naturally in the method consists in extending the signal using anti-symmetric mirror boundary conditions. It is defined as $s_k = -s_{-k-1}$, and $s_{N-1+k} = -s_{N-2-k}$, for $k = 0, 1, \dots, N-1$, repeated on the further extensions of the signal. As can readily be verified, this is equivalent to requiring that $s(-x) = -s(x-1)$ and that $s(x)$ is $(2N-2)$-periodic. Note however that, unlike the symmetric extension, this one cannot be applied to arbitrary signals, as it requires that $s_{N-1} = -s_{N-2}$. Actually, an anti-symmetric signal is always zero mean. Once again, it is sufficient to specify the anti-symmetry around the origin; the anti-symmetry on the other end is propagated automatically through the periodization process.

Propagation of the boundary conditions

Any shift-invariant operator preserves the $(2N-2)$ periodicity, but not necessarily the symmetry. We therefore need to investigate how Δ and Δ^{-1} propagate symmetric and anti-symmetric boundary conditions. We need also to correct for the fact that the considered periodic signals are not necessarily zero mean.

The finite differences operator Δ inverses symmetry. Specifically, it transforms anti-symmetric into symmetric boundary conditions and symmetric into *shifted* anti-symmetric boundary conditions. The following theorem claims that Δ^{-1} has a similar behavior.

Theorem 1 *The operator Δ^{-1} transforms symmetries according to*

- Symmetric input
  If $s(x) = s(-x)$ and s has a zero mean, then $u = \Delta^{-1} * s$ satisfies $u(-x) = -u(x-1)$. Thus, if s satisfies symmetric boundary conditions, then $\Delta^{-1} * s$ satisfies anti-symmetric boundary conditions.

- Anti-symmetric input
  If $s(-x) = -s(x-1)$ and s has a zero mean, then $u = \Delta^{-1} * s$ satisfies $u(-x) = u(x-2)$. Thus, if s satisfies anti-symmetric boundary conditions, then $\delta(x-1) * \Delta^{-1} * s$ satisfies symmetric boundary conditions.

Proof: It is sufficient to prove the property for a finitely supported signal $s(x)$ that satisfies the zero-mean property $\sum_k s(x-k) = 0$, because of the definition of Δ^{-1} for periodic signals.

- *Symmetric input*
  We have

$$\begin{aligned} u(-x) &= \sum_{k\geq 0} s(-x-k) && \text{definition of } \Delta^{-1} \\ &= \sum_{k\geq 0} s(x+k) && \text{symmetry property} \\ &= -\sum_{k\leq -1} s(x+k) && \text{zero-mean property} \\ &= -u(x-1). \end{aligned}$$

- *Anti-symmetric input*
  We have

$$\begin{aligned} u(-x) &= \sum_{k\geq 0} s(-x-k) && \text{definition of } \Delta^{-1} \\ &= \sum_{k\geq 0} -s(x+k-1) && \text{antisymmetry property} \\ &= \sum_{k\leq -1} s(x+k-1) && \text{zero-mean property} \\ &= u(x-2). \end{aligned}$$

■

We have defined our inverse finite differences operator for finitely supported signals, or zero mean periodic signals. We will now show how to deal with non

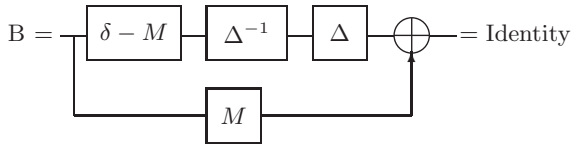


Figure 4.5: Identity diagram for symmetric boundary input signals

zero mean P-periodic signals in our algorithm. Let us define the moving average filter

$$M(x) = \frac{1}{P} \sum_{k=0}^{P-1} \delta(x-k).$$

We can then consider the identity block diagram in Figure 4.5, which holds for symmetric boundary input signals. The key idea is that the output of $\boxed{\delta - M}$ is a signal of zero mean, while that of $\boxed{M}$ is a constant, whenever the input is P-periodic.

Since the finite differences operator kills constant signals, we can write the equivalence shown in Figure 4.6(a) for an input with symmetric boundary conditions. Furthermore, in order to implement the boundary conditions as specified in Theorem 1, it is necessary to define a "symmetric" version of Δ^{-1}, Δ_{S}^{-1}, for symmetric inputs and an "anti-symmetric" version, Δ_{A}^{-1}, for anti-symmetric inputs as shown in Figure 4.6(b). Note that Δ_{A}^{-1} is simply Δ^{-1} (delayed by one sample) because an anti-symmetric input is of zero mean. That is why, the filter $\boxed{\delta - M}$ disappear at this stage. This is iterated (n_1+1)-times to yield the equivalence in Figure 4.6(c). Note that the alternation between symmetric and anti-symmetric boundary conditions adds a delay of $\lfloor \frac{n_1+1}{2} \rfloor$.

Thus, in practice, we will use an alternation of Δ_{S}^{-1} and Δ_{A}^{-1} instead of $\Delta^{-(n_1+1)}$, adding the appropriate delay. This ensures that the boundary conditions are correctly propagated throughout. This modification is necessary for the behavior of the algorithm to be fully consistent; in particular, this ensures that for n odd and for an integer scaling (including $a = 1$) the method is fully reversible with no boundary artifacts.

4.3 Generalization of the method

We now show that the projection method can also be implemented exactly for a more general class of piecewise polynomial functions.

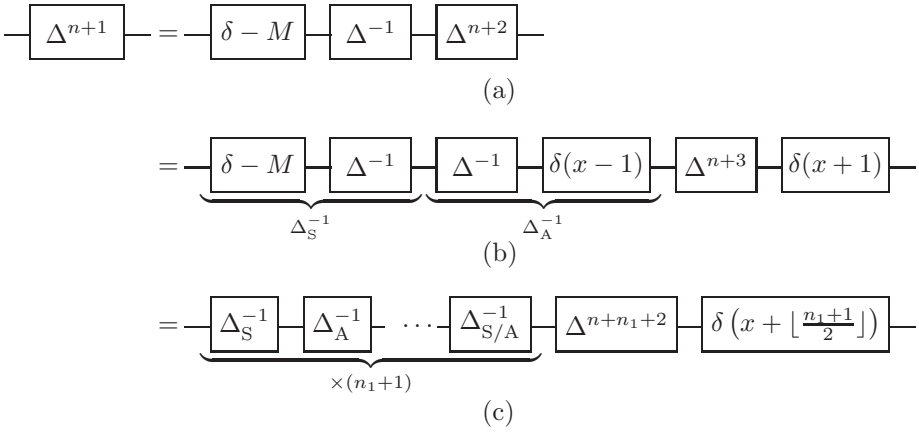


Figure 4.6: Equivalences for symmetric boundary input signals. (a) First iteration: We apply the identiy block diagram in Figure 4.5. (b) Second iteration: We define a "symmetric" version of Δ^{-1}, Δ_{S}^{-1}, for symmetric inputs and an "anti-symmetric" version, Δ_{A}^{-1}, for anti-symmetric inputs. (c) (n_1+1)-iteration of the process: The alternation between symmetric and anti-symmetric boundary conditions adds a delay of $\lfloor \frac{n_1+1}{2} \rfloor$.

4.3.1 Linear combinations of shifted B-splines

We consider the case where the basis functions are linear combinations of shifted splines

$$\psi(x) = \sum_i \alpha_i \beta^n(x + h_i) \longleftrightarrow \hat{\psi}(\omega) = F(\omega)\hat{\beta}^n(\omega), \tag{4.2}$$

with $F(\omega) = \sum_i \alpha_i e^{j\omega h_i}$. These functions are piecewise polynomial. However, they are not necessarily splines and the knots are not necessarily uniformly spaced. Then, the functions $\varphi(x)$, $\varphi_1(x)$ and $\varphi_2(x)$ used in our algorithm depicted in Figure 4.2, are defined by $(\alpha_i, h_i) = (\alpha_{0,i}, h_{0,i})$, $(\alpha_{1,i}, h_{1,i})$ and $(\alpha_{2,i}, h_{2,i})$, respectively; moreover, we have $a_{12}(k) = q_k^{-1} = \sum_{i,j} \alpha_{1,i}\alpha_{2,j}\beta^{n_1+n_2+1}(k + h_{1,i} - h_{2,j})$.

If we follow the same process as in Section 4.2.1, we end up with a diagram

similar to Figure 4.3-e with a kernel that is now

$$\phi(x) = a^{n_1+1} \sum_{i,j} \alpha_{0,i}\alpha_{1,j}\beta^{n+n_1+1}(x+\tau_{i,j})$$

where $\tau_{i,j} = h_{0,i} - \frac{h_{1,j}}{a} + b + \frac{n_1+1}{2}\left(\frac{1}{a}-1\right)$.

4.3.2 Linear combinations of B-spline derivatives

We are interested in this class of functions because, as was explained in Chapter 2, the MOMS functions that minimize the support N_φ for a given order L, are linear combinations of B-spline derivatives

$$\varphi(x) = \sum_{k=0}^{L-1} \gamma_k \frac{d^k}{dx^k}\beta^{L-k}(x). \tag{4.3}$$

We get a simple expression for $\varphi(x)$ in terms of one-sided power functions by using the relation between splines of different degrees,

$$\varphi(x) = \sum_{i=0}^{n} \gamma_i D^i \beta^n(x) = \sum_{i=0}^{n} \gamma_i \Delta^i * \beta^{n-i}\left(x+\frac{i}{2}\right) \tag{4.4}$$

$$= \Delta^{n+1} * \sum_{i=0}^{n} \gamma_i \frac{x_+^{n-i}}{(n-i)!} * \delta\left(x+\frac{n+1}{2}\right). \tag{4.5}$$

We now select $\varphi(x)$, $\varphi_1(x)$ and $\varphi_2(x)$ to be linear combinations of B-spline derivatives of degree n, n_1 and n_2, respectively, with coefficients $\gamma_{0,i}$, $\gamma_{1,i}$ and $\gamma_{2,i}$. With this particular choice, we get

$$a_{12}(k) = \langle \varphi_1(x), \varphi_2(x-k)\rangle = \sum_{j=0}^{n_1}\gamma_{1,j}\sum_{i=0}^{n_2}\gamma_{2,i}\Delta^{i+j} * \beta^{n_1+n_2-i-j+1}\left(k+\frac{i+j}{2}\right),$$

and the final scheme is the same as Figure 4.3-e with

$$\phi(x) = \sum_{i=0}^{n}\gamma_{0,i}\sum_{j=0}^{n_1}\gamma_{1,j}a^{n_1-j+1}\Delta^{i+j} * \beta^{n+n_1-i-j+1}(x+\tau_{i,j})$$

and $\tau_{i,j} = \frac{n_1+1}{2}\left(\frac{1}{a}-1\right) + \frac{i+j}{2} + b$.

Interestingly, the generalized scheme has the same computational cost as the B-spline algorithm. The basis functions are polynomials of the same degree as the corresponding splines; they have the same support and the recursive prefilters have the same degree.

Note that this particular setting also constitutes a limit case of the previous one. Specifically, we can approximate the derivative operator using finite differences and make the sampling step h tend to 0:

$$\varphi(x) = \sum_{i=0}^{n} \gamma_i D^i \beta^n(x) = \lim_{h \to 0} \sum_{i=0}^{n} \gamma_i \frac{1}{h^i} \Delta_h^i * \beta^n(x)$$

$$\begin{aligned} \hat{\varphi}(\omega) &= \left(\sum_{i=0}^{n} \gamma_i (j\omega)^i \right) \hat{\beta}^n(w) \simeq \sum_{i=0}^{n} \gamma_i \left(\frac{-1 + e^{j\omega h}}{h} \right)^i \hat{\beta}^n(\omega) \\ &= \sum_{k=0}^{n} \alpha_k e^{j\omega h_k} \hat{\beta}^n(\omega) = F(\omega) \hat{\beta}^n(\omega), \end{aligned}$$

with $\alpha_k = \sum_{i \geq k} \binom{k}{i} \gamma_i \frac{1}{h^i} (-1)^{i-k}$ and $h_k = kh$.

4.3.3 Extensions of the method

In principle, our algorithm can also be extended to higher dimensions and to non-separable geometric affine operators. One may get the intuition for this extension by stressing the key feature of our setting: the function $\varphi_1(x)$ which appears in the general projection-based scheme of Figure 4.2 is built using shifted versions of functions—the one-sided power functions x_+^n—that are easily exchanged through the geometric transformation—the scaling operator.

The idea is thus to choose a function ρ that can easily be exchanged through the geometric transformation, and to require that φ_1 belongs to the space generated by the uniform shifts of ρ; in our algorithm, $\rho(x) = x_+^n$.

For instance, if we wanted to implement rotations and scalings of an N-dimensional digital signal, we could define $\rho(\mathrm{x}) = \|\mathrm{x}\|^n$, that is, a radial basis function. This radial basis function can be localized using a digital filter Δ_ρ; that is to say, $\Delta_\rho * \rho$ defines a function that has some appropriate decay as $\|\mathrm{x}\| \to \infty$. In our algorithm, this localization filter is simply the finite differences operator Δ^{n+1}, which transforms x_+^n into a B-spline of degree n. We would finally need to compute the convolution of the localized radial basis function

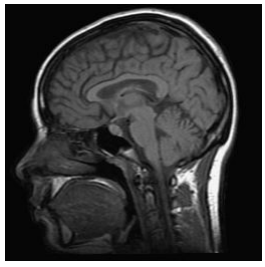


Figure 4.7: Original Magnetic Resonance (MR) image.

with φ to get the function ϕ shown in Figure 4.3-e. The disadvantage of this generalization, however, is that the underlying basis functions are no longer compactly supported.

4.4 Experimental results

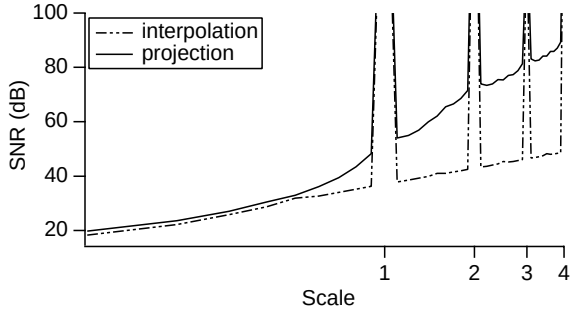


Figure 4.8: Projection versus interpolation method for linear splines

We used a series of back and forth experiments to evaluate and compare the various resizing algorithms. A test image—the MR scan (Figure 4.7)—is scaled by a factor of a and then set back to its initial size using the reverse transformation (scaling by a factor of a^{-1}) with the same algorithm. The loss of information is measured by the relative mean square difference between the approximation and the initial digital image, expressed in decibels (dB). The experiment is repeated for many scaling factors and the peak signal-to-noise

ratio (SNR) is represented as a function of the scale in a logarithmic plot. Scale factors smaller than 1 correspond to image reduction, while scale factors larger than 1 represent enlargement. Obviously, most information is lost in the reduction step, not in the enlargement one. Note, however, that magnification is not fully reversible unless the zooming factor is an integer.

4.4.1 Least-squares versus interpolation

Our first goal was to compare the performance of our projection algorithm with the more standard interpolation method that fits the image with a spline of the same degree and then resamples it at the required rate. The detailed results for $n = 1$ (linear splines) are given in Figure 4.8. It is clear from this plot that the least-squares method outperforms the standard one (bilinear interpolation), even though the underlying model is the same in both cases.

The visual improvement can be substantial, as illustrated in Figure 4.10 and 4.11. We observe that the small-scale details are much better preserved with our optimal approach (see Figure 4.10 and 4.11) and the contrast is enhanced because of the reduction of aliasing. Figure 4.9 illustrates the reduction of blocking artifacts of the projection method with respect to the standard one.

Interestingly, the projection method also provides some improvement for image magnification. For $a > 1$, the gain is of the order of 20 dB. The distance between the two curves when $a > 1$ reflects the differences between the leading asymptotic constants: In the orthogonal projection case C_L is small and in the interpolation case C_L^{int} is larger (see Section 2.3). We also note that the error curve exhibits peaks at the integers, which simply reflects the fact that the signal is preserved exactly for integer zooming factors. In this particular case, the interpolation and projection methods are equivalent because the corresponding spline spaces are nested (which implies that the projection error is zero). This is a property that holds for all B-splines of odd degrees, but not for the O-MOMS; for splines of even degrees, it is only true for odd magnification factors.

The superiority of the least-squares method is also apparent for the other interpolation models as shown in Figure 4.12. This graph displays the relative SNR improvement of least-squares versus interpolation for splines of degree 0, 1, and 3, as well as the cubic O-MOMS. For small scaling factors ($a < 0.4$), the improvement is typically better than 2 dB, irrespective of the model used. The fundamental reason for the lesser performance of standard interpolation is aliasing. The effect is more pronounced for large reduction factors or when the image contains a lot of high frequency information.

Another visual example is provided in Figure 4.13. Here, we observe a

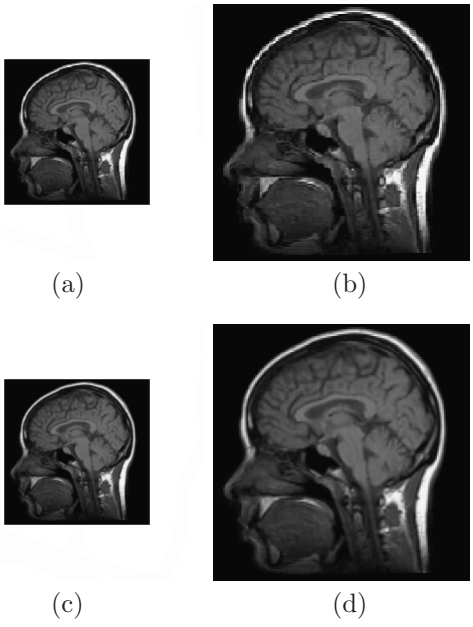


Figure 4.9: Example of image reduction by a factor $a = \sqrt{\pi}$ using splines of degree 0. Notice that the projection method reduces blocking artifacts: (a) Reduced image using standard method; (b) Enlarged version of the image (a) (SNR=25.8 dB); (c) Reduced image using orthogonal projection; (d) Enlarged version of the image (c) (SNR=30.7 dB).

substantial improvement in the perceptual quality of the projection method over the standard one in rescaling text. The interpolation model used was cubic.

4.4.2 Comparison of basis functions

Now that we have established the superiority of the projection method, it is interesting to compare the various basis functions. In particular, we are interested in evaluating the effect of the order parameter L. For this comparison, we use as our reference the least-squares method with cubic splines, which corresponds to the error graph in Figure 4.14-a. The relative performance comparison of the various models is shown in Figure 4.14-b. As expected, the SNR improves as

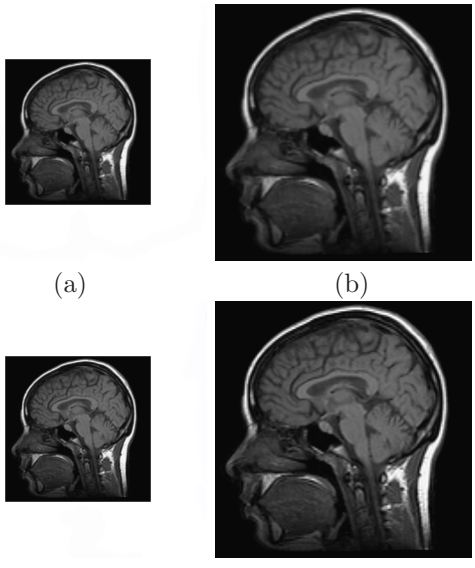
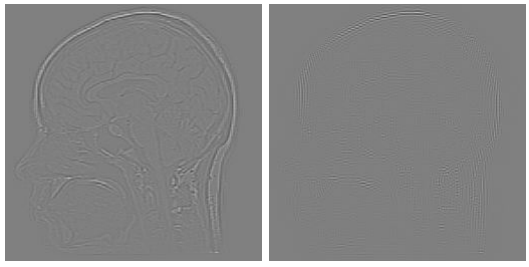


Figure 4.10: Same experiment as Figure 4.9, but with linear splines. Notice the aliasing reduction from (b) to (d) (better contrast of the features). See also the difference images in Figure 4.11: (a) Reduced image using standard method; (b) Enlarged version of the image (a) (SNR=31.9 dB); (c) Reduced image using orthogonal projection; (d) Enlarged version of the image (c) (SNR=35 dB).

the order of the spline increases. For small reduction factors ($a < 0.5$), cubic splines perform 1.0 dB better than linear splines, and 2.5 dB better than the piecewise constant model ($n = 0$). For large scale factors, this difference gets magnified. If we now compare the O-MOMS and cubic B-splines, which have the same support ($W = 4$) and the same order ($L = 4$), we find that the former offer slightly better performance across all scales (+0.15 dB at small scales), which confirms their optimality.

We also compared the methods when the image is only shifted forward by a factor b and backwards by the same factor without resizing. Figure 4.15 shows the results. We observe that the O-MOMS give the best value in terms of SNR, 1 dB over cubic splines, while the linear splines (resp., piecewise constant) are



(a) (b)

Figure 4.11: Difference between the original and the enlarged version of the reduced image obtained in Figure 4.10 with the linear spline resizing methods: (a) Standard method; (b) Orthogonal projection.

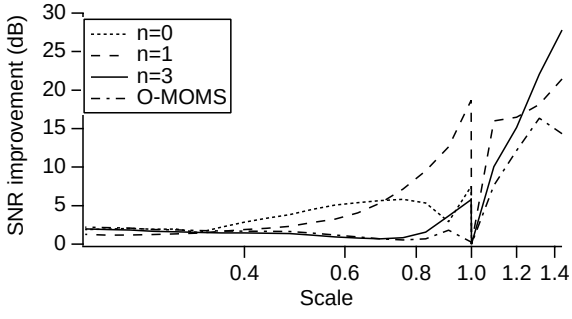


Figure 4.12: Least-squares versus various interpolation models.

10 dB (resp., 25 dB) below the cubic ones. Thus, it appears that higher order correlates with improved shift-invariance, in accordance with the theoretical findings in [20].

4.4.3 Oblique versus orthogonal projection

When we pick the analysis degree n_1 different from n, our method implements an oblique projection instead of an orthogonal one. In Figure 4.16, we see that such an oblique projection only brings a slight degradation of 0.4 dB when $n_1 = 0$ and 0.15 dB when $n_1 = 1$ compared to the orthogonal scheme, with the advantage of a lesser computational complexity ($O(n_1 + n + 1)$ instead of

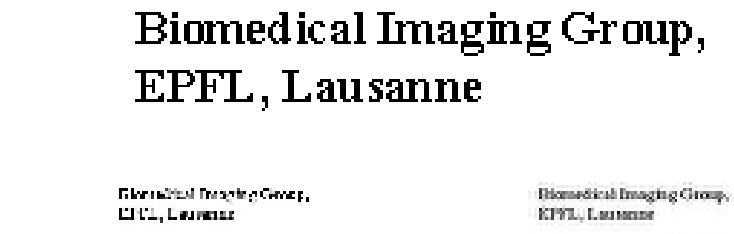


Figure 4.13: Resizing method using cubic splines applied to text with a scale factor 0.33; Top—Original text; Bottom—Reduced image: (left) using standard method; (right) using orthogonal projection.

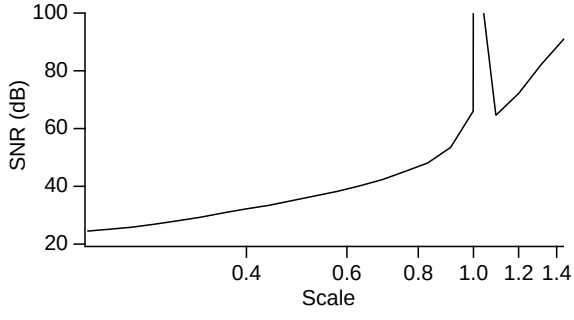
$O(2n+1)$). These results are consistent with the theory developed in [124].

4.5 Summary

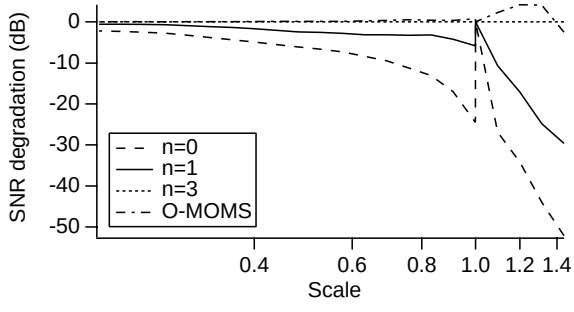
In this chapter, we have generalized Lee *et al.'s* [73] method for image resizing using both oblique and orthogonal projections. We have demonstrated that the new method outperforms the standard interpolation techniques. It is especially advantageous for image reduction because of the built-in anti-aliasing mechanism. An attractive property of the present implementation is that the complexity per output point does not depend on the scaling factor. Our resizing algorithm works for arbitrary scaling factors (image magnification or reduction) thanks to the underlying mathematical formalism worked out in Chapter 3. We believe that it should be useful in applications where image quality is a key concern.

The formulation of the resizing problem that has been presented is rather general. By varying some key parameters, we switch between optimal least-squares solution, oblique projection and interpolation. We have also described algorithmic solutions for basis functions other than B-splines, the most notable example being the O-MOMS.

In the second half of the present chapter, we have addressed the correct handling of the boundaries when implementing the discrete finite differences operators. The edge-handling mechanism is simple and based in periodization and two types of boundary conditions: symmetric and anti-symmetric.

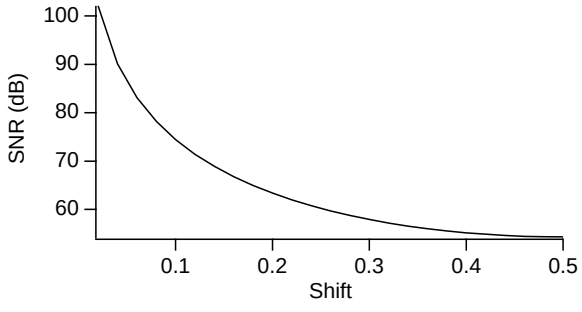


(a)

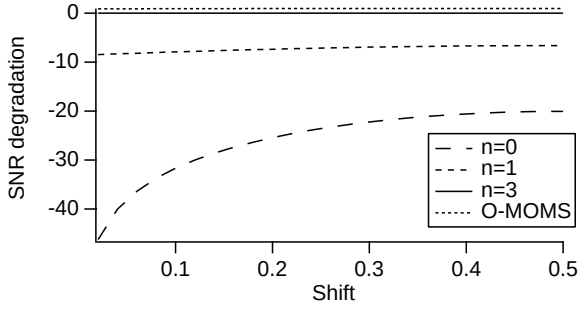


(b)

Figure 4.14: SNR measures for the least-squares projection method; (a) Cubic splines; (b) Comparison of cubic splines with $n = 1, 0$ degree splines and O-MOMS.

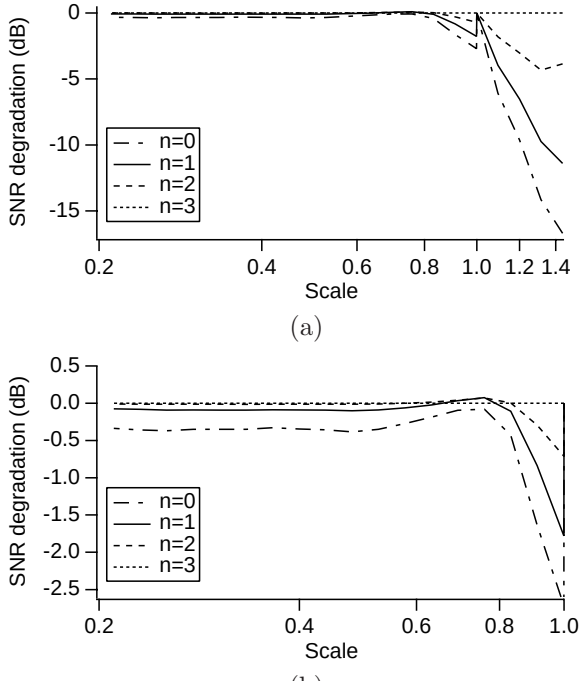


(a)



(b)

Figure 4.15: SNR measures for the shift variations using the projection method. (a) Cubic splines; (b) Comparison in the performance of different splines with the cubic ones.



(a)

(b)

Figure 4.16: Loss in performance by using oblique projection instead of least-squares. (a) Full scale range [0.2 : 1.4]; (b) Reduced scale range [0.2:1.0] to magnify the difference at low scale factors.

Chapter 5

Non-uniform to Uniform Grid Conversion

Non-uniform to uniform grid conversion is another approximation problem where our L_2-projection approach onto a spline space offers an advantage.

The interest of performing the non-uniform to uniform grid conversion comes from its wide range of applications. Examples of these are format conversion for display or processing purposes and curve resampling in computer graphics [101]. Another promising application field is signal reconstruction from non-uniformly distributed samples coming from domains such as metrology, biomedical [86, 102, 107], or robotics. Commonly used methods for that reconstruction includes iterative algorithms, modified Fourier transforms and interpolation.

In this chapter, we solve the problem of approximating a non-uniform spline $f(x)$ with a uniform spline of the form $\tilde{f}(x) = \sum_k c_k \beta^n(x-k)$, where $\beta^n(x)$ is the non-centered B-spline of degree-n.

The simplest approach, as we know from Chapter 2, would be to sample $f(x)$ uniformly and then compute the spline that interpolates these samples. Its main drawback is the introduction of aliasing when the sampling density is not high enough.

Here, we propose as alternative to minimize the L_2-approximation error between the non-uniform spline and its uniform representation. We have derived a closed form solution for this least-squares approximation problem. Our implementation is computationally exact and works for arbitrary sampling rates.

As we treat the problem in the continuous domain, we avoid the possible

ill-posedness of the discrete matrix formulation [2].

We restrict ourselves to 1-D signals only. It is conceivable too, to extend the method to non-uniform N-D interpolation using thin-plate splines, for instance.

5.1 Philosophy of the approach

We follow the same philosophy as for the resizing (Section 4.1) and calculate the orthogonal projection of $f(x)$ onto the uniform causal spline space $V^n = \text{span}\{\beta^n(x-k)\}_{k\in\mathbb{Z}}$ where β^n is a noncentered B-spline.

The corresponding projection formula is

$$\tilde{f}(x) = Pf(x) \quad = \quad \sum_k (c * q)_k \beta^n(x-k), \tag{5.1}$$

where

$$c_k \quad = \quad \langle f(x), \beta^{n_1}(x-k)\rangle. \tag{5.2}$$

and where q is the IIR spline filter with z-transform $Q(z) = \frac{1}{\sum_k \beta^{n+n_1+1}(k) z^k}$ that has a fast recursive implementation as described in [128]. Thus, the only remaining difficulty as was pointed out in Chapter 3 is to compute (5.2); i.e., the inner products of the non-uniform spline $f(x)$ with B-splines of degree n_1.

5.2 Non-uniform splines

We call $B_i^n(x)$ the non-uniform B-spline of degree $n > 0$ associated with the knots $\lambda_i, \ldots, \lambda_{i+n+1}$; it is given by [34]

$$B_i^n(x) = (n+1)[\lambda_i, \ldots, \lambda_{i+n+1}](x-\lambda)_+^n$$

with $[\lambda_i, \ldots, \lambda_{n+i}]f(\lambda)$ the divided differences of order n of the polynomial $f(\lambda)$ defined as

$$[\lambda_i, \ldots, \lambda_{n+i}]f(\lambda) = \sum_{k=i}^{n+i} \frac{f(\lambda_k)}{\prod\limits_{l=i, l\neq k}^{n+i} (\lambda_l - \lambda_k)}.$$

The support of B_i^n is finite. More precisely, $B_i^n(x) = 0$ if $x \notin [\lambda_i, \lambda_{i+n+1}]$. This is due to the localization properties of the divided differences with respect

to polynomials. We construct the B-splines of degree n from the one of degree $(n-1)$ by applying De Boor's recursion formula [23],

$$B_i^n(x) = \left(\frac{n+1}{n}\right)\left[\frac{x-\lambda_i}{\lambda_{i+n+1}-\lambda_i}B_i^{n-1}(x) + \frac{\lambda_{i+n+1}-x}{\lambda_{i+n+1}-\lambda_i}B_{i+1}^{n-1}(x)\right]. \quad (5.3)$$

By using these basis functions, we can represent any non-uniform spline as

$$f(x) = \sum_i a_i B_i^n(x).$$

5.3 Non-uniform to uniform grid conversion algorithm

5.3.1 Derivation of the algorithm

As we said before, the difficulty of the method is to compute the inner product of the non-uniform spline with a uniform one. The key formula was derived in Chapter 3 from the uniform B-spline definition

$$c_k = \langle f, \beta^{n_1}(x-k)\rangle = \Delta^{n_1+1} * D^{-(n_1+1)} f(k).$$

In other words, if we have the uniform samples of $D^{-(n_1+1)}f(k)$, we can compute the B-spline coefficients c_k by simple digital filtering. Thus, it is of critical importance to be able to compute the (n_1+1)-fold integral of a non-uniform spline. The recursive formula that we use for the calculation is derived from recursion (5.3) and the integration formula for a non-uniform spline [24]. Specifically,

$$D^{-1}B_i^{n-1}(x) = \sum_{j\geq i} \frac{\lambda_{j+n+1}-\lambda_j}{n+1} B_j^n(x). \quad (5.4)$$

Note that $D^{-1}B_i^{n-1}(x) \in \text{span}\{B_j^n\}_{k\in\mathbb{Z}}$.

5.3.2 Practical implementation

The main steps of the algorithm are summarized below and shown in Figure 5.1.

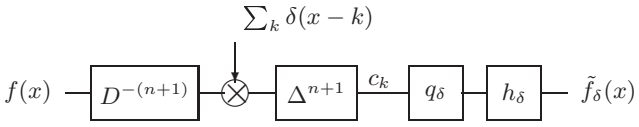


Figure 5.1: General scheme for the method (orthogonal projection)

- **Interpolation**

  Either the spline $f(x) = \sum_i a_i B_i^n(x)$ is known, or it is specified by non-uniform samples of $f(x)$ at knot positions λ_k. In this latter case, the coefficients a_i of the spline interpolant are the solution of a band-diagonal system of equations [23].

- **Integration**

  In Sections 5.2 and 5.3.1, we have presented all the tools required to calculate the (n_1+1)-fold integral of $f(x)$. We write

  $$D^{-1}f(x) = \sum_i a_i D^{-1} B_i^n(x) = \sum_i a_i^{(-1)} B_i^{n+1}(x)$$

  with

  $$a_i^{(-1)} = \frac{\lambda_{i+n+2} - \lambda_i}{n+2} \sum_{j \geq i} a_j \tag{5.5}$$

  We then define

  $$I(x) = D^{-(n_1+1)} f(x) = \sum_i a_i^{-(n_1+1)} B_i^{n+n_1+1}(x)$$

  The basic idea is that the (n_1+1)-fold integral of a non-uniform spline of degree n is a non-uniform spline of degree $(n+n_1+1)$.

  The coefficients $a_i^{-(n_1+1)}$ are pre-computed by recursive application of (5.5). Likewise, we may also update the B-spline basis functions by using De Boor's recursion formula (5.3).

- **Sampling**

  The next step is the resampling of the above integral at the integers

  $$I(k) \quad = \quad D^{-(n_1+1)} f(x) \mid_{x=k} = \sum_i a_i^{-(n_1+1)} B^{n_1+n+1}(k) \,.$$

- **Digital Filtering**

  At this stage we apply $(n_1 + 1)$ centered finite differences corresponding to the filter with z-transform $\Delta^{n_1+1} \leftrightarrow (1 - z)^{n_1+1}$ and the IIR filter $b^{-(n+n_1+1)}$ specified in [125]. Finally, we get the output from the system by postfiltering with $h_k = \beta^n(k)$, the sampled version of the synthesis function.

5.4 Experimental results

We will now compare the grid conversion method which uses interpolation and resampling with our projection method to get some indication of the kind of improvement that can be achieved.

Our first test signal is a non-uniform cubic spline that approximates the line number 9 of Figure 5.2 using 69 knots (see Figure 5.3). When we just resample it at uniform sample locations and interpolate the result with a uniform spline, we observe that the spline curve is constrained to pass through the sampling points as shown in Figure 5.4(a) (Peak SNR= 26.321 dB). When the projection method is applied, the resulting uniform spline curve tries to adjust itself to minimize the difference with the non-uniform spline one as shown in Figure 5.4(b) (Peak SNR= 27.395 dB). It leads to a better performance in terms of SNR (see Figure 5.5(a)). The improvement for a ratio of uniform to non-uniform knot number between 1 and 2 is 1 dB on average as illustrated in Figure 5.5(b). The sampling points for the uniform interpolation are the same in both cases.

As a second example, we represent the word *here* with a non-uniform cubic spline using 22 control points as illustrated in Figure 5.6. A curve in the x—y plane can be represented in terms of an arbitrary parameter t as $r(t) = (x(t), y(t))$. We have chosen here to specify the curve by a reduced number of control points $\lambda_k = (\lambda_{x,k}, \lambda_{y,k})$ and to parametrize to each pair with the arc-length t_k of the curve between the origin and the knot (curvilinear axis). The reason for doing so is that such a cubic spline provides the minimum curvature interpolant of these control points [65]. We obtain a continuous representation $r(t)$ by interpolation of each component $(\lambda_{x,k}, t_k)$ and $(\lambda_{y,k}, t_k)$ by a non-uniform cubic spline resulting in

$$x(t) = \sum_k d_k B_k^3(t) \text{ and } y(t) = \sum_k g_k B_k^3(t)$$

where d_k and g_k are the coefficients of the non-uniform spline representation.

We process each non-uniform spline component independently using either the standard or the projection method. Note that a uniform knot distribution means here equal length of the curve between adjacent points.

Figure 5.7 represents the uniform cubic splines recovered when using 26 uniform knots. The output from the projection method is barely readable (peak SNR= 34.820 dB) while the output from the standard method is less (peak SNR= 32.934 dB).

The increase of the SNR as a function of the ratio of uniform to non-uniform knot number is shown in Figure 5.8(a). It is seen from Figure 5.8(b) that the projection method is superior to the standard one by 2 to 0.5 dB with a gain that decreases as the ratio increases.

5.5 Summary

We have presented a refined tool to convert non-uniform splines into uniform ones. The approach is efficient computationally; its complexity per output point is constant and independent of the knot spacing of the input signal. It is easy to check that the cost of an oblique projection into V^n perpendicular to V^{n_1} is approximately equivalent to a (non-uniform) spline interpolation of degree $(n + n_1 + 1)$.

Experimental results show that our method outperforms standard interpolation; this to be expected because our solution is optimal in the least-squares sense. Moreover, the implicit analog prefiltering step in our approximation formula (5.1) reduces aliasing.

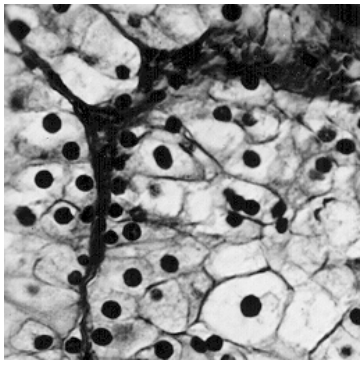


Figure 5.2: Original image.

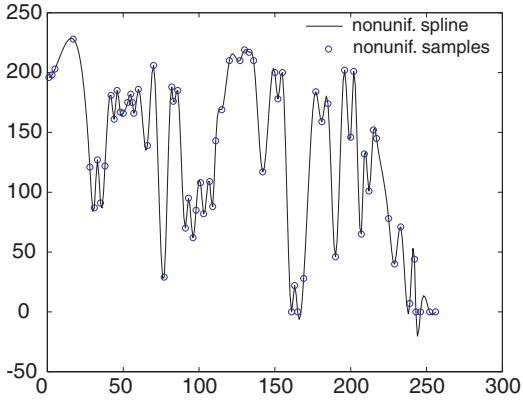
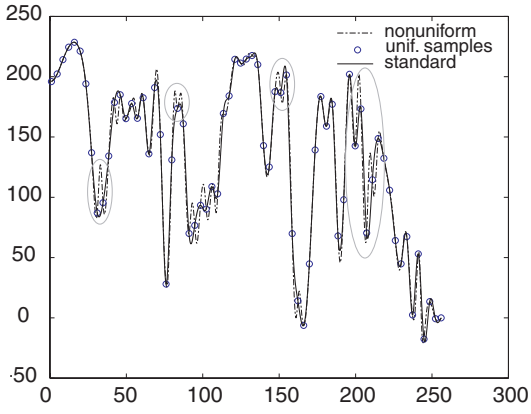
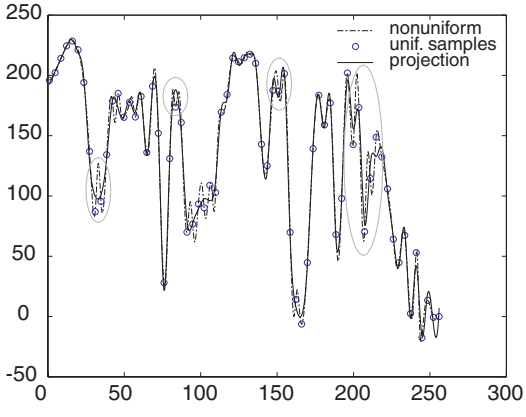


Figure 5.3: Test signal which is a non-uniform cubic spline that approximates the line number 9 of Figure 5.2. The circles correspond to the 69 non-uniform knot positions that describe the non-uniform spline.

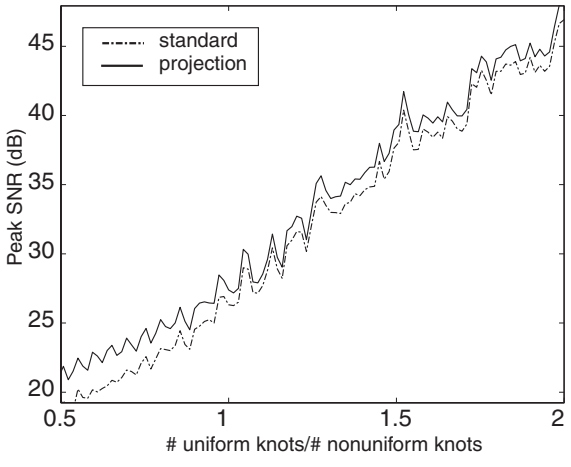


(a)

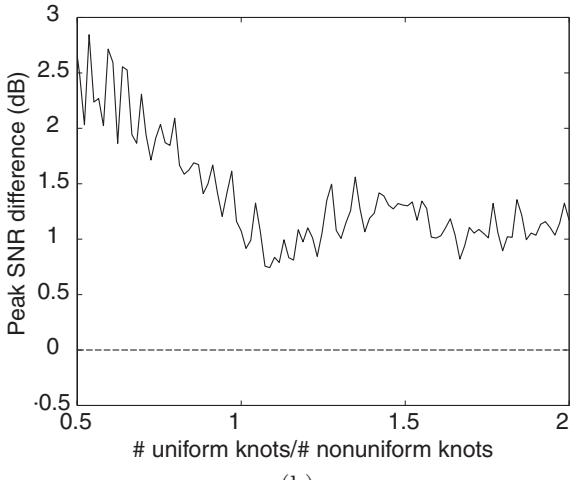


(b)

Figure 5.4: Non-uniform to uniform grid conversion. The number of uniform sampling points is the same as the number of non-uniform control points (69). (a) Standard method: The uniform spline is forced to pass through the positions defined by the sampling points (Peak SNR= 26.321 dB); (b) Projection method: The smoothing effect of the projection the minimizes the L_2-error(Peak SNR= 27.395 dB). Look at the areas inside the circles to observe the main differences.



(a)



(b)

Figure 5.5: Comparison of the peak SNR for the standard and the projection method for different ratios of uniform to non-uniform knot number. The input non-uniform spline is the one given in Figure 5.3. (a) Peak SNR values. (b) Peak SNR difference.

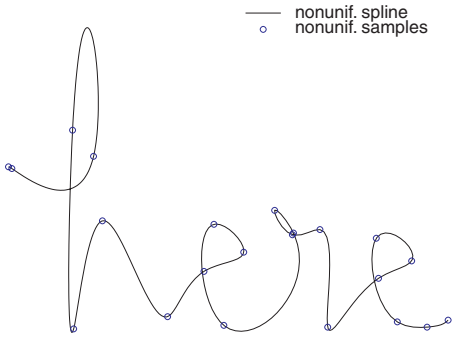
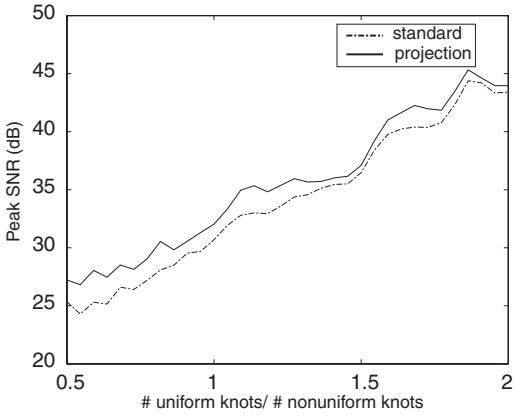
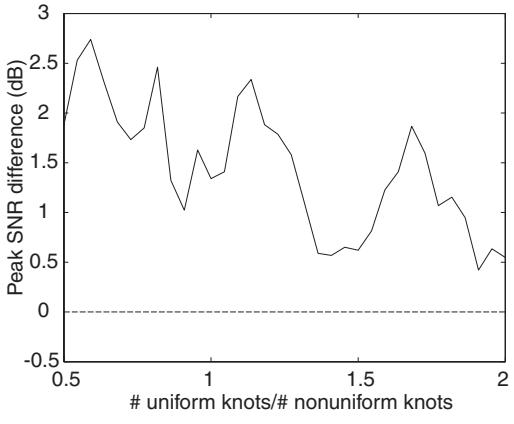


Figure 5.6: Non-uniform cubic spline constructed with the help of the 22 marked control points.

Figure 5.7: Non-uniform to uniform grid conversion. The number of uniform sampling points is 26. (a) Standard method: Peak SNR= 32.934 dB; (b) Projection method: Peak SNR= 34.820 dB. The word is readable here.



(a)



(b)

Figure 5.8: Comparison of the peak SNR for the standard and the projection method for different ratios of uniform to non-uniform knot number. The input non-uniform spline is the one given in Figure 5.6. (a) Peak SNR values. (b) Peak SNR difference.

Chapter 6

ℓ_p-Multiresolution Analysis

Multiresolution analysis is a simple yet very powerful concept which goes back to the pioneering works of Rosenfeld [103] and Burt and Adelson [27]. Instead of a fixed size pixel array, one considers a hierarchical image description at multiple resolution levels; typically, a series of fine-to-coarse approximations which are stored in a pyramid data structure. Such pyramids are extremely useful for speeding up computations. In fact, there are multi-scale versions of most image processing algorithms. The main advantages of multi-scale processing are the following:

- Computational speed: Since there are much fewer pixels at the coarser levels of the pyramid, iterative algorithms that switch between resolution levels require less computation and have faster convergence.
- Spatial resolution adaptation: Many image processing algorithms operate on very localized neighborhoods and it makes good sense to adapt the resolution in an optimal fashion. This is especially true with iterative schemes which proceed by successive refinement—here the resolution should be linked to the step size of the algorithm.
- Increased robustness: In the context of iterative algorithms, the smoothing effect of the pyramid reduces the likelihood of getting trapped in local extrema.
- Analogies can be made with the hierarchical organization of the human primary visual cortex.

Multiresolution analysis also plays a central role in the theory of the wavelet transform, which provides a non-redundant representation of images across scales. Here too, the applications in image processing and computer vision are numerous, and often very successful [43, 78, 79].

One of the earliest and most popular example of pyramid is due to Burt and Adelson [27]. Their Gaussian filtering, however, produces excessive smoothing, which leads to some loss of image details. Higher-quality image approximation can be obtained by designing a reduction filter that is optimum in the least-squares sense, or by using the lowpass branch of a wavelet decomposition algorithm [1, 77]. Another option is to use spline pyramids that minimize either the ℓ_2 or the L_2-approximation error [128, 130]. These latter representations are especially attractive for continuous/discrete multiscale processing. The nice feature of these pyramids is that they can all be implemented using a combination of filters and sampling rate converters. Of course, the critical aspect here is filter design—a standard requirement is the biorthogonality of the reduction and expansion operators [134]. Unfortunately, simplicity also comes at a price and these pyramids suffer from limitations that are inherent to linear methods; in particular, edge blurring (when the smoothing is to too strong; e.g., the Gaussian pyramid), aliasing (when it is not enough), and ringing artifacts (when the filters have a sharp cut-off). Thus, the selection of a suitable multiresolution model is essentially a question of compromise: higher order spline or wavelet approximations generally yield better energy compaction but they also give rise to larger Gibbs oscillations as the functions become more and more bandlimited [7].

An attractive alternative to linear pyramids is to go non-linear. Several authors have proposed to replace the linear pyramid filters by non-linear ones including the median and morphological operators [37, 51, 83, 111, 122]. However, these so-called morphological pyramids are generally not meant to provide a continuous/discrete representation. Non-linear filters can also introduce distortions that make the reduced images visually unpleasant.

In this thesis, we will pursue another approach and introduce spline pyramids that are optimal for ℓ_p-norms. Note that the choice of a spline model in this context is equivalent to specifying the expansion mechanism—i.e., polynomial spline interpolation. Thus, the challenge is to come up with a corresponding reduction operator that produces visually pleasant results without aliasing and with minimal ringing artifacts. The information lost by the reduction operator should be as little as possible. Normally, this corresponds to difference images which are sparse, i.e., that contain a lot of small-valued elements.

This chapter is organized as follows. In Section 6.1, we present a Banach-space formulation of the problem and prove that the solution is well defined. In Section 6.2, we propose a digital filtering-based procedure that computes the solution iteratively. The computational overhead of the iterative approach is small, and we expect the generation of the pyramid to account for only a very small part of the total effort in a typical multiscale algorithm. In Sections 6.3.1 and 6.3.2, we compare the approximations obtained for different p's and orders of the approximation functions, respectively.

6.1 Multiresolution subspaces of ℓ_p

In this section, we present the theoretical basis of our method. We describe the continuous/discrete model chosen in the light of the arguments presented in Chapter 2 and show that our approximation problem has a well-defined solution.

6.1.1 Definitions and notation

The $\|\cdot\|_{\ell_p}$ norm of a sequence $c = \{c_k\}_{k\in\mathbb{Z}}$ is defined as

$$\|c\|_{\ell_p} = \left(\sum_{k\in\mathbb{Z}} |c_k|^p\right)^{\frac{1}{p}} \tag{6.1}$$

with $1 \leq p < \infty$ and the special case $\|c\|_\infty = \max_{k\in\mathbb{Z}} |c_k|$.

The z-transform of a signal $s(k), k \in \mathbb{Z}$ is denoted by

$$S(z) = \sum_{k\in\mathbb{Z}} s(k) z^{-k}$$

If we make $z = e^{j2\pi f}$, we recover the discrete Fourier transform.

The symbol $\downarrow N$ denotes the downsampling operator by the integer factor N; it is defined as

$$s_{\downarrow N}(k) = s(Nk), \quad \forall k \in \mathbb{Z}$$

The dual operator $\uparrow N$ represents upsampling by the integer factor N

$$s_{\uparrow N}(k) = \begin{cases} s(\frac{k}{N}), & \text{if } N \text{ divides } k, \\ 0, & \text{elsewhere.} \end{cases}$$

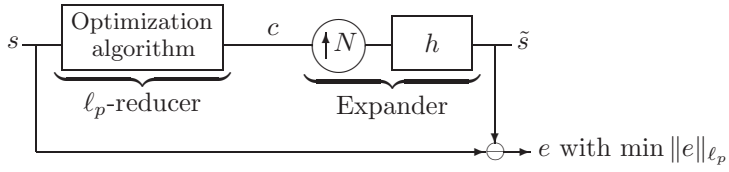


Figure 6.1: Reduction/expansion system for an integer scaling factor N. Reduction: the signal is reduced by a factor N so as to minimize the ℓ_p-norm of the error. Expander: upsampling and filtering, as specified by the approximation model (see equation (6.2)).

6.1.2 Approximation signal model

Our signal model (cf. (6.2) below) follows the philosophy presented in Chapter 2. For simplicity, we will present the theory in $1D$. The extension to multiple dimensions is straightforward through the use of tensor product basis functions. The use of a separable model implies that the expansion mechanism is separable as well; the reduction mechanism, on the other hand, will not be separable unless we are dealing with the classical case $p = 2$ (least squares approximation).

Specifically, we choose to represent all signals in terms of shifted basis functions, which are typically sampled B-splines. A discrete signal, e.g. $s(k)$, will always denote the samples on the finest grid. Its coarser level approximation $\tilde{s}(k)$ at resolution N will use basis functions $\varphi_l(k)$ that are obtained by translation and sampling at the integers of a continuous basis function $\varphi(x) \in L_2(\mathbb{R})$ dilated by N, i.e., $h_k = \varphi\left(\frac{k}{N}\right)$. We make this choice because we want the interpolated version of our approximated signal to belong to the space $\text{span}\{\varphi\left(\frac{x}{N} - l\right)\}_{l\in\mathbb{Z}}$, i.e., $\tilde{s}(x) = \sum_{l\in\mathbb{Z}} c_l \varphi\left(\frac{x}{N} - l\right)$, while its samples $\tilde{s}(k)$ belong to $\text{span}\{h\left(k - Nl\right)\}_{l\in\mathbb{Z}}$:

$$\tilde{s}(k) = \sum_{l\in\mathbb{Z}} c_l h_N(k - Nl) = [c]_{\uparrow N} * h_N(k) \tag{6.2}$$

This yields a consistent discrete/continuous signal representation. As we already know from Chapter 2, the advantage of this joint model is the possibility of applying continuously-defined operators commonly used in image processing such as derivatives or geometrical transformations.

In other words, $\tilde{s}(k)$ is entirely specified by its coefficients c_l. These are the quantities that are stored in the pyramid; at each level there is exactly one such

number per node. The template h_N should be interpreted as an expansion (or interpolation) filter which maps the coefficients sequence c_l (coarse level of the pyramid) to the finest resolution level on which the signals are defined (cf. the right hand side of Figure 6.1). We are using the subscript N in h_N to indicate that the expansion filter depends on N (typically, a spline interpolator with an expansion factor N). In the sequel, we will sometimes leave out this dependence to simplify the notation. The corresponding approximation space is

$$V_N = \left\{ \tilde{s}(k) = \sum_{l \in \mathbb{Z}} c_l h_N(k - Nl) : c \in \ell_p \right\} \tag{6.3}$$

It is clearly convex and N-integer shift-invariant; i.e., $s(k) \in V_N$ iff $s(k+N) \in V_N$.

For our formulation, it is essential that V_N be a closed subspace of ℓ_p to ensure a well-defined solution of our approximation problem. This will be the case if $\{h(k - Nl)\}_{k \in \mathbb{Z}}$ forms a p-stable (or p-Riesz) basis,

$$\forall c \in \ell_p,\ A \cdot \|c\|_{\ell_p} \leq \left\| \sum_{l \in \mathbb{Z}} c_l h(k - Nl) \right\|_{\ell_p} \leq B \cdot \|c\|_{\ell_p} \tag{6.4}$$

with $0 < A, B < \infty$.

This norm equivalence implies that $h \in \ell_p$ (by letting $c_l = \delta_l$) and that ℓ_p and V_N are isomorphic Banach spaces.

6.1.3 Riesz-basis theorem in Banach spaces

This above condition is ensured by the following theorem:

Theorem 2 *If $h \in \ell_1$ and $\{h(k - lN)\}_{l \in \mathbb{Z}}$ is a p_0-Riesz basis for some $1 \leq p_0 \leq \infty$ then it is also a p-Riesz basis for $1 \leq p \leq \infty$.*

Consequently, if h is in ℓ_1 and generates a Riesz basis in the conventional ℓ_2-sense, then it is automatically also p-stable for any p. The following result by Aldroubi *et al.* [4] gives a simple way to check if $h \in l_1$ generates a Riesz basis or not.

Theorem 3 *$\{h(k - lN)\}_{l \in \mathbb{Z}}$ is a ℓ_2-Riesz basis if and only if*

$$0 < \alpha \leq \sum_{i=0}^{N-1} \left| H\left(e^{\frac{j2\pi(f-i)}{N}} \right) \right|^2 \leq \beta < +\infty.$$

We will now proceed and prove Theorem 2 using a technique that was inspired by the work of Aldroubi *et al.* [3] who considered L_p-Riesz basis in a continuous framework. The proof makes use of two classical results

Proposition 1 (Young's Inequality) *If $b \in \ell_1$ and $a \in \ell_p$ then $\|a * b\|_{\ell_p} \leq \|b\|_{\ell_1} \cdot \|a\|_{\ell_p}$.*

Lemma 1 (Wiener's Lemma) *Let $a \in \ell_1$ with $A(e^{j2\pi f}) \neq 0 \ \forall f$, then $(a)^{-1} \leftrightarrow \frac{1}{A(e^{j2\pi f})}$ is in ℓ_1 as well.*

These are used to establish the following:

Lemma 2 *Let $h \in \ell_1$ generate a N-shift-invariant ℓ_2-Riesz basis. Then, its ℓ_2-dual $\mathring{h}$ defined by $\mathring{h} = (a_h^{-1})_{\uparrow N} * h$ with $a_h(l) = \langle h(k), h(k - lN)\rangle = (h * h^T)_{\downarrow N}$, is in ℓ_1 as well.*

Proof:
Thanks to Young's inequality, we have that $a_h \in \ell_1$ because

$$\|a_h\|_{\ell_1} = \|(h * h^T)_{\downarrow N}\|_{\ell_1} \leq \|h * h^T\|_{\ell_1} \leq \|h\|_{\ell_1}^2 < +\infty.$$

Since $\{h(k - lN)\}_{l \in \mathbb{Z}}$ is a ℓ_2-Riesz basis, the autocorrelation function

$$\alpha \leq \underbrace{\sum_{i=0}^{N-1} \left| H\left(e^{\frac{2\pi(f-i)}{N}}\right)\right|^2}_{A_h(e^{j2\pi f})} \leq \beta$$

is positive definite (cf. Theorem 3). The conditions of Wiener's lemma are met; thus, $a_h \in \ell_1$ implies that a_h^{-1} belongs to ℓ_1 as well. We show that $\mathring{h} \in \ell_1$ by using the Young's inequality and the fact that upsampling does not change the value of the norm.

$$\|\mathring{h}\|_{\ell_1} = \|(a_h^{-1})_{\uparrow N} * h\|_{\ell_1} \leq \|a_h^{-1}\|_{\ell_1} \cdot \|h\|_{\ell_1}$$

■

To prove Theorem 2, we need to establish upper and lower bounds in the norm equivalence:

$$\forall c \in \ell_p, A \cdot \|c\|_{\ell_p} \leq \|s\|_{\ell_p} = \|c_{\uparrow N} * h\|_{\ell_p} \leq B \cdot \|c\|_{\ell_p}$$

The upper bound is easily localized using Young's Inequality and the fact that $\|c_{\uparrow N}\|_{\ell_p} = \|c\|_{\ell_p}$.

$$\|c_{\uparrow N} * h\|_{\ell_p} \leq \|c\|_{\ell_p} \underbrace{\|h\|_{\ell_1}}_{B}$$

To determine a lower bound we will work with the dual filter $\mathring{h}$ which is in ℓ_1 as well, as a consequence of Lemma 2. Since $\mathring{h}$ and h are biorthogonal, we have that $\forall s \in V_N$, $s(k) = \sum_{l \in \mathbb{Z}} \underbrace{\langle s, \mathring{h}(k - lN) \rangle}_{c_l} h(k - lN)$.

$$\|c\|_{\ell_p} = \|[s * \mathring{h}^T]_{\downarrow N}\|_{\ell_p} \leq \|s * \mathring{h}^T\|_{\ell_p} \leq \underbrace{\|\mathring{h}\|_{\ell_1}}_{A^{-1}} \|s\|_{\ell_p}$$

with $s = c_{\uparrow N} * h$. So, we have found A such as $A \cdot \|c\|_{\ell_p} \leq \|c_{\uparrow N} * h\|_{\ell_p}$.

One advantage of working with B-splines or spline-related basis function (spline interpolator) is that these conditions will be satisfied irrespective of the reduction factor N. Indeed, Aldroubi *et al.* have shown that the B-splines and cardinal splines with step size N generate a Riesz basis of ℓ_2 [8]. Since these splines are also in ℓ_1, they satisfy the conditions on h to have V_N a closed subspace of ℓ_p.

6.1.4 Projection theorem in Banach spaces

Given the discrete signal $s \in \ell_p$, we would like to find the approximation $\tilde{s} \in V_N \subset \ell_p$ that minimizes the error $\|s - \tilde{s}\|_{\ell_p}$. The projection theorem in Banach spaces (i.e., ℓ_p) [70] states that, since V_N is a convex closed subspace of ℓ_p for any $s \in \ell_p$, there exists $\tilde{s} \in V_N$ such that

$$\|s - \tilde{s}\|_{\ell_p} = d(s, V_N) = \inf_{s_V \in V_N} \|s - s_V\|_{\ell_p} \tag{6.5}$$

Thus, $\tilde{s}$ is the best approximation of s in V_N, in the ℓ_p-sense. We denote $\tilde{s} = P_{V_N} s$. For $1 < p < \infty$, $\tilde{s}$ is unique. Unicity is lost for $p = 1$ and $p = \infty$; nevertheless, all the minima are global, ensuring that the solution to our

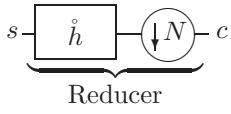


Figure 6.2: Optimal reducer for ℓ_2-norms: antialiasing filter and downsampler.

approximation problem is well-defined. Thus, our initial problem of calculating the minimum error approximation translates into calculating the coefficients c_l in (6.2) that describe the projection $\tilde{s}$.

6.2 Optimal approximation

In this section, we refer to the state-of-the-art algorithm to calculate ℓ_2-projections. We then turn to more general ℓ_p-projections and present a novel iterative algorithm based on digital filtering.

6.2.1 Optimal approximation in ℓ_2

We will start by presenting the solution of Aldroubi *et al.* [4]. For $p = 2$, our space is a Hilbert space; i.e., a Banach space with an inner product. In that case, the calculation of $P_{V_N} s$ takes the simpler form

$$\tilde{s}(k) = P_{V_N} s(k) = \sum_{l \in \mathbb{Z}} \langle s, \mathring{h}(k - Nl) \rangle h(k - Nl) \tag{6.6}$$

where $\mathring{h}$ is the (unique) dual function of h; i.e., $\mathring{h} \in V_N$ and $\langle \mathring{h}(k - Nl), h(n - Nl) \rangle = \delta(k - n)$ (biorthonormality).

The coefficients c_l of the orthogonal projection of the input signal $s \in \ell_1$ onto V_N are given by

$$c_l = [s * \mathring{h}^T]_{\downarrow N}(l) \tag{6.7}$$

where

$$\mathring{H}(z) = \frac{NH(z^{-1})}{\sum_{k=0}^{N-1} H(e^{j\frac{2\pi k}{N}} z) H(e^{j\frac{2\pi k}{N}} z^{-1})}.$$

The corresponding reduction/expansion digital filtering system is shown in Figure 6.1. In this particular case, the reduction is implemented via a prefilter $\mathring{h}$ followed by a downsampler as shown in Figure 6.2. Note that the Riesz condition ensures that the filter $\mathring{h}$ exists and is always well-defined.

6.2.2 Optimal approximation in ℓ_p

Now, we deal with the general case of finding an optimal ℓ_p-approximation. The difficulty of working in Banach spaces is the lack of an inner product. Practically, this means that the solution cannot be computed by a one step linear algorithm. In this section, we develop an iterative optimization procedure that takes advantage of linear filtering and of the calculation of first and second order derivatives.

The ℓ_p-norm of the approximation error $e = s - \tilde{s}$ is a convex function of the coefficients c_l, which ensures that its local minima are also global. This is because of the constitutive definition of a norm (esp., triangle inequality and semilinearity) and because the error e depends linearly on c. The consequence is that a gradient-based optimization algorithm with adaptive steps will always converge to the global minimum. However, since for $p = 1$ the norm of the error e is only piecewise differentiable, we must be prepared to encounter some difficulties (slower convergence) as p gets close to 1.

Theoretical derivation of the optimization algorithm

To speed up convergence, we propose a robust optimization algorithm, the formulation of which is Hessian-based. The idea behind it is to optimize the coefficients c_l each in turn and to express the norm of the error as a second order polynomial which is easily minimized. The update formula for the vector of coefficients is then derived.

Mathematically, we justify our algorithm as follows: If we fix an index l_0 in the expression for $\tilde{s}$, we have

$$\tilde{s}(k) = \sum_{l\in\mathbb{Z}} c_l h(k - Nl) = c_{l_0} h(k - Nl_0) + \underbrace{\sum_{l\neq l_0} c_l h(k - Nl)}_{\eta(k)} \tag{6.8}$$

Then, we rewrite the norm of the error as

$$\begin{aligned} \|e\|_{\ell_p}^p &= \|s-\tilde{s}\|_{\ell_p}^p \\ &= \sum_{k\in\mathbb{Z}} |e(k)|^{p-2}(s(k)-c_{l_0}h(k-Nl_0)-\eta(k))^2 \end{aligned} \tag{6.9}$$

We minimize the last expression as a function of c_{l_0} considering $|e(k)|^{p-2}$ as independent of c_{l_0}, in order to get the update formula. Thus, if we know $c^{(i)} = (\dots, c_{-1}^{(i)}, c_0^{(i)}, \dots . c_{l_0}^{(i)}, \dots)$, we obtain the update vector of coefficients $\Delta c^{(i+1)}$ by calculating for each index l_0

$$\Delta c_{l_0}^{(i)} = c_{l_0}^{(i+1)} - c_{l_0}^{(i)} = -\frac{\sum_{k\in\mathbb{Z}} |e(k)|^{p-2} e(k) h(k-Nl_0)}{\sum_{k\in\mathbb{Z}} |e(k)|^{p-2} h^2(k-Nl_0)} \tag{6.10}$$

and then $\tilde{s}^{(i+1)} = \tilde{s}^{(i)} + \sum_{l\in\mathbb{Z}} \Delta c_l^{(i)} h(k-Nl)$.

We now show that this algorithm can also be interpreted as a gradient-based or quasi-Newton search procedure. The partial derivative of the norm of the error $\|e\|_{\ell_p}^p$ with respect to c_l is

$$\frac{\partial \|e\|_{\ell_p}^p}{\partial c_l} = -\sum_{k\in\mathbb{Z}} g_1[e(k)] h(k-Nl)$$

with $g_1(x) = p|x|^{p-2}x$. The second order partial derivative is

$$\frac{\partial^2 \|e\|_{\ell_p}^p}{\partial c_l \partial c_n} = \sum_{k\in\mathbb{Z}} g_2[e(k)] h(k-Nl) h(k-Nn)$$

with $g_2(x) = p(p-1)|x|^{p-2}$; these define the entries of the (infinite dimensional) Hessian matrix H.

The update formula for the usual Hessian algorithm [89] takes the form $\Delta c^{(i)} = -H^{-1}\nabla e$ where ∇e is the gradient (vector of partial derivatives) and H is the Hessian (matrix of second order partial derivatives). Here, the Hessian matrix is essentially diagonal dominant because $h(k)$ is decaying away from the origin. It is therefore legitimate to use the following simplified update formula

$$\Delta c^{(i)} = -\lambda\ (\mathrm{diag} H)^{-1}\nabla e, \tag{6.11}$$

where we have also introduced a step size λ.

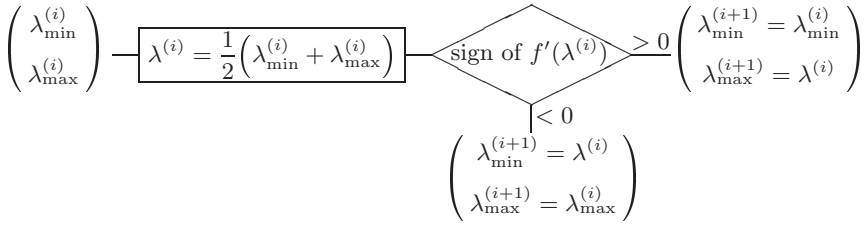


Figure 6.3: One iteration of the algorithm that evaluates the optimal λ.

If we make $\lambda = p - 1$, we have the equivalence with formula (6.10). In the following of the chapter, we will call "Hessian fixed" the algorithm described by (6.11) with $\lambda = p - 1$.

We will see that the advantage of the Hessian over the gradient-based methods is its efficiency, especially when p gets close to 1. It costs slightly more per iteration because the diagonal of the Hessian has to be evaluated in addition to the gradient, fortunately, we can compute it efficiently, using filtering and downsampling.

Implementation of the optimization algorithm

We describe now the modular structure of the optimization algorithm designed to calculate the coefficients c_l of the ℓ_p-approximation signal. The implementation uses two reduce operations (Figure 6.4) followed by an expander (Figure 6.1). The update vector $\Delta c^{(i)}$ is obtained from the error in three steps. First, gradient estimation (Figure 6.4-upper branch), then, inverse of the diagonal of the Hessian estimation (Figure 6.4-lower branch). Those are finally combined and multiplied by the step size λ to provide the update vector $\Delta c^{(i)}$. The diagram of Figure 6.1 shows how to recompute the error at the given iteration.

The value of the step size λ in (6.11) can be made optimal in the sense of minimizing the error as much as possible at each step. The idea is to remark that we are minimizing

$$f(\lambda) = \left\| e^{(i)} - \lambda \sum_k u(l)h(k - Nl) \right\|_{\ell_p} \tag{6.12}$$

with $u = (\mathrm{diag}H)^{-1}\nabla e$ according to (6.11). In practice, we estimate an upper ($\lambda_{\max}$) and lower ($\lambda_{\min}$) bound for λ. This value is optimized by using a line

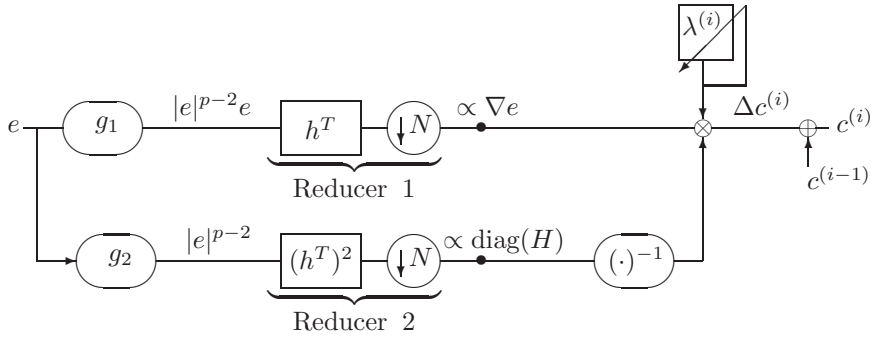


Figure 6.4: Optimal reducer for ℓ_p-norms: Reducer 1: Gradient estimation. Reducer 2: Diagonal of the Hessian estimation. The value of λ that minimizes the error at each iteration is calculated using a line search algorithm. The filters are reversed versions of the original ones: $h^T(k) = h(-k)$

Table 6.1: Gradient algorithm: Average (and standard deviation) number of iterations of the line search algorithm (Figure 6.3)

Gradient	mean $\pm$ stdv
ℓ_3-optimal	3.36 ± 1.82
ℓ_2-optimal	1.0 ± 0.0
$\ell_{1.2}$-optimal	9.83 ± 1.27
$\ell_{1.05}$-optimal	8.74 ± 1.84

search algorithm that reduces by two the length of the interval $[\lambda_{\min}, \lambda_{\max}]$ at each step. Figure 6.3 describes one iteration of the algorithm. The search for the optimal λ is made acceptable in terms of computational overhead by choosing initial bounds quite close to the optimal; i.e., those calculated in the preceding iteration. If $f'(\lambda_{\min}^{(0)}) < 0$ and $f'(\lambda_{\max}^{(0)}) > 0$ then the convergence is ensured by the convexity of $f(\lambda)$. Note that for $p = 2$ we have an exact formula to calculate λ optimal that amounts to minimizing a second order polynomial.

We observe from Tables 6.1 and 6.2 that the line search algorithm requires approximately $10 - 14$ iterations when $p < 2$ in order to yield an optimal step

Table 6.2: Hessian algorithm: Average (and standard deviation) number of iterations of the line search algorithm (Figure 6.3)

Hessian	mean ± stdv
ℓ_3-optimal	10.92 ± 0.89
ℓ_2-optimal	1.0 ± 0.0
$\ell_{1.2}$-optimal	14.41 ± 1.43
$\ell_{1.05}$-optimal	14.29 ± 1.29

size λ. The test image is the one in Figure 6.6-top. Cubic splines are chosen for the interpolation and the scale is reduced by a factor of two.

Figure 6.5 represents a typical example of convergence of the algorithm for the minimization of the ℓ_p-approximation error. The test signal and the parameters are the same as for the example given above. Each graph shows the decrease of the criterion (increase of the $\mathrm{SNR}_{\ell_p} = -20 \log \left(\frac{\|s-\tilde{s}\|_{\ell_p}}{\|s\|_{\ell_p}} \right)$ dB), measured as a function of the number of iterations for different values of p. The results are shown in dBs. Four variants of our algorithm are compared: Either gradient or Hessian-based with the parameter λ calculated in an optimal fashion; gradient-based with λ fixed ($\lambda = 0.00045$ for $p = 3$, $\lambda = 0.12$ for $p = 2$, $\lambda = 1.0$ for $p = 1.2$ and $\lambda = 2.0$ for $p = 1.05$); Hessian-based as given by equation (6.11) with $\lambda = p - 1$. We observe in Figure 6.5 (a) that for high values of p ($p = 3.0$ in this case), the performances of the Hessian-based and gradient optimal algorithms are very similar. The convergence of the gradient-based algorithms with λ fixed is worse. Figure 6.5 (b) is a special case as we deal with the convergence of the least-squares approximations. Here, the diagonal of the Hessian is constant and independent of the input, which implies that the Hessian and gradient-based algorithms that use the same strategy for determining λ are equivalents. On the other hand, we observe a slightly faster increase of the SNR_{ℓ_p} for the algorithms with λ optimal over the other ones. Figures 6.5 (c) and (d) demonstrate the behavior of the algorithms with p's close to 1 ($p = 1.2$ and $p = 1.05$, respectively). Here, the algorithms with λ optimal converges in less iterations than their counterparts in a global sense. We observe a slower decrease of the error in the first iterations of the gradient method with λ fixed, due to the conservative step size we have chosen to ensure convergence when we are close to the solution. The Hessian algorithm moves as fast as the version that uses the λ optimal at the beginning but afterwards; it gets slower as the

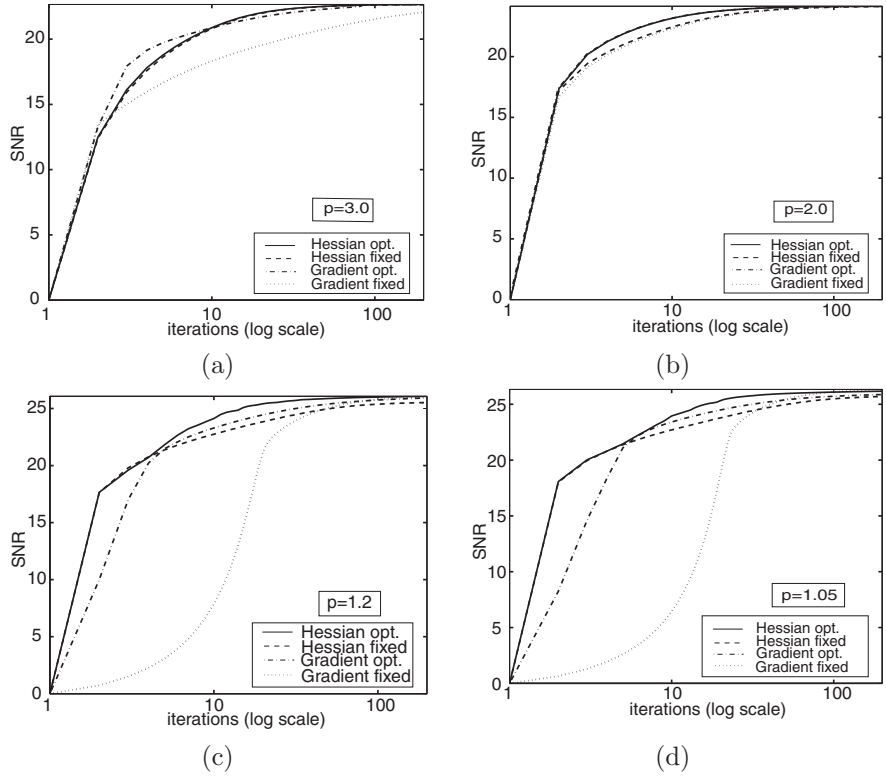


Figure 6.5: Convergence of the algorithm. Each graph illustrates the decrease of the criteria (i.e., the increase of the SNR_{ℓ_p}) as a function of the number of iterations for a different value of p. The axes are logarithmic. Results for: (a) $p = 3.0$ and $\lambda_{\mathrm{fixed}} = 0.00045$ (b) $p = 2.0$ and $\lambda_{\mathrm{fixed}} = 0.12$ (c) $p = 1.2$ and $\lambda_{\mathrm{fixed}} = 1.0$ (d) $p = 1.05$ and $\lambda_{\mathrm{fixed}} = 2.0$.

diagonal of the Hessian gets larger for p close to 1 (when $p = 1$ the denominator of (6.10) becomes $\sum_{k\in\mathbb{Z}} |e(k)|^{-1} h^2(k - Nl)$).

In conclusion, we recommend the Hessian-based, fixed step-size algorithm when $p \geq 2$ as it gives almost the same performance as the one that uses line search. For p close to 1, the algorithms that take advantage of λ optimal are more robust at the price of an added computational cost. Note that each

iteration has complexity comparable to that of the ℓ_2-projection. What makes the ℓ_p-algorithm computationally more expensive is the number of iterations required for reaching the solution. Based on the results in Figure 6.5, we may conclude that $10 - 20$ iterations are necessary. However, we found empirically that if instead of $c^{(0)} = 0$ we start with the ℓ_2-solution, we save of the order of 10 iterations for $p < 2$.

6.2.3 Generation of image pyramids

If the basis functions φ used to specify h satisfy a two-scale relation, then the dyadic multiresolution for the linear case has the nestedness property of the vector spaces: $\ldots \subset V_{2^i} \subset V_{2^{i-1}} \subset \ldots \subset V_1$ [4, 98]. The discrete wavelet transform that minimizes the L_2-norm exploits this nestedness by computing the projection at one scale from the previous finer approximation. This hierarchical approach is not appropriate here. In principle, one should always go back to the finest scale to compute the coarse level approximations because of the nonlinear structure of the reduction operator.

In Figure 6.6, we illustrate this distinction. We have generated the pyramid on the left hand side using the optimal $\ell_{1.2}$ approach (we take the finest resolution image as initial image to calculate all coarser approximations). The pyramid on the right hand side is suboptimal in the sense that each coarse-level approximation is computed from the previous finer level approximation. The reconstruction error is measured by the SNR_{ℓ_p} as defined before (here, $p = 1.2$). It is evident that the error of the approximation at level 1 is the same in both cases (26.11 dB) as we start from the same image. As can be expected, the error is slightly larger for the suboptimal (18.74 dB (level 2) and 14.53 dB (level 3)) than for optimal (18.99 dB (level 2) and 14.77 dB (level 3)) pyramid. In practice, these differences are not significant and it is quite justifiable to use the step-wise suboptimal approach to minimize computation. Nevertheless, in the following tests we did not use the recursive downsampling approach but the direct one.

6.3 Experimental results

6.3.1 Comparison of approximations for different p's

In this section, we characterize the pyramid decompositions for different values of p.

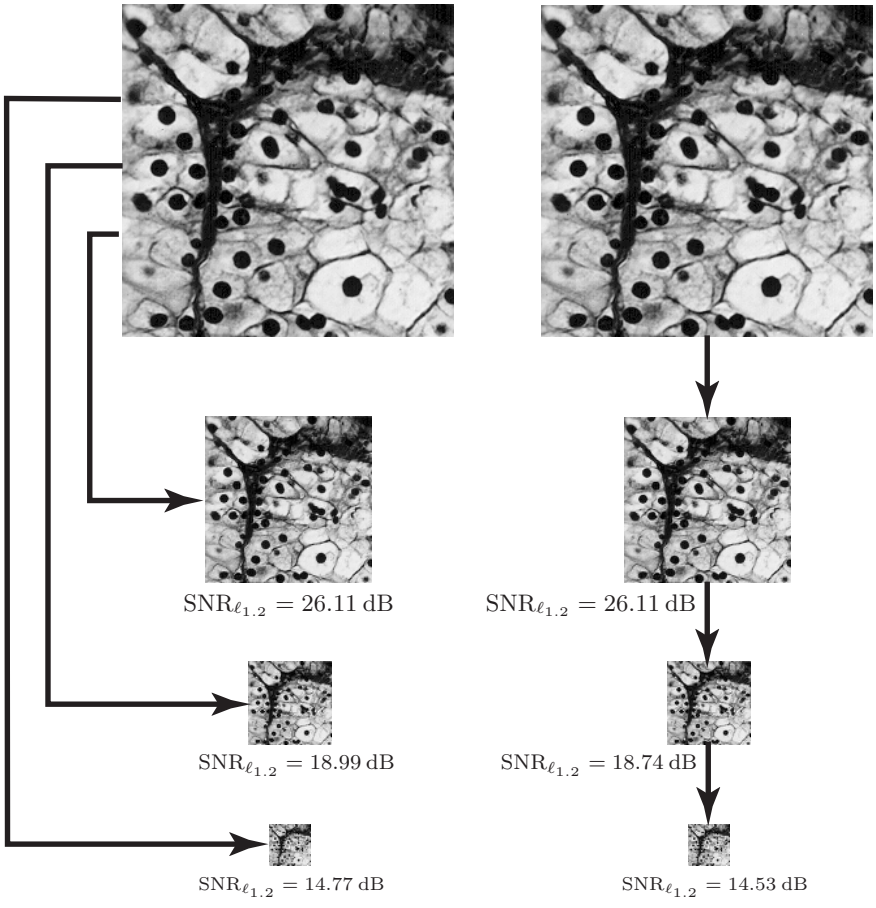


Figure 6.6: Optimal versus suboptimal $\ell_{1,2}$ pyramid.

We observe two effects: The ringing is reduced and the histogram of the error gets more concentrated as p gets closer to 1.

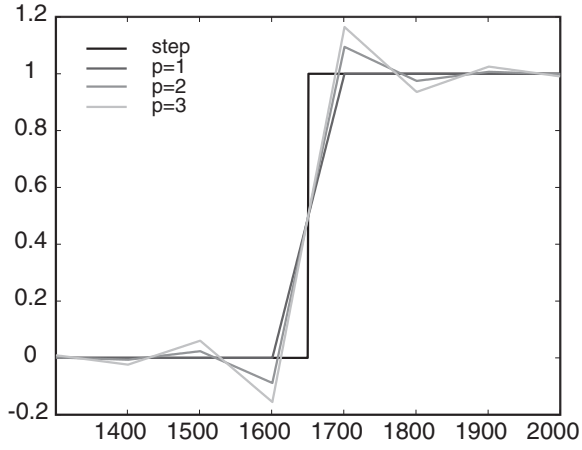
Ringing

First, we describe the Gibbs phenomenon in 1D. When a signal $s(x)$ is discontinuous, its Fourier transform decreases slowly at high frequencies. The Gibbs oscillations are created by the removal of the highest frequency components by a low-pass filtering (i.e., when calculating a least-squares approximation), or by an equivalent system (i.e., an optimal ℓ_p-reducer). The Gibbs oscillations have an amplitude proportional to the discontinuity jump and independent of the reduction factor N. This artifact is frequently observed in least-squares approximations and is well documented in the literature. We now present an experiment which aims at quantifying the effect in the context of ℓ_p-approximations.

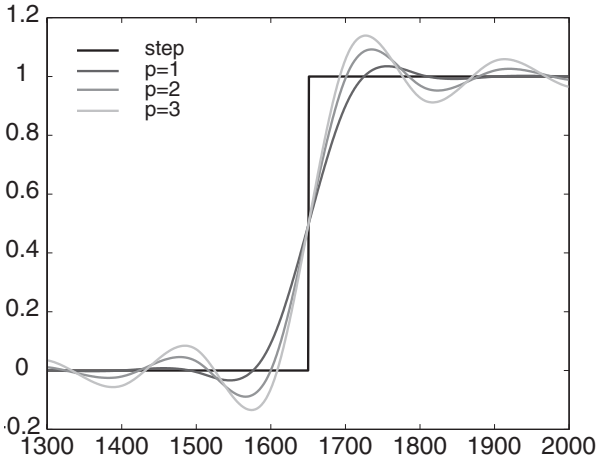
We have chosen a 1D signal that is a single discontinuity: a step function. Its amplitude is unity and made of 3200 samples. We have reduced it by a factor $N = 100$ using different values of p. Figure 6.7 shows the original step and the signals interpolated back to the original size. We have represented only the interesting region around the discontinuity. The basis functions are linear (Figure 6.7 (a)) and cubic B-splines (Figure 6.7 (b)). Observe that there is almost no Gibbs phenomenon for $p = 1$, and that the oscillations get more pronounced as p increases.

We have chosen to quantify the Gibbs phenomenon by the amplitude of the overshoot given by $\|\tilde{s}\|_{\ell_\infty}$. Figure 6.8 shows the increase of $\|\tilde{s}\|_{\ell_\infty}$ in percent with p for linear and cubic spline approximations. We note that the overshoot for linear spline approximations is comparable to that of cubic spline approximations.

To illustrate the effects of the reduction of the ringing for images, we show in Figure 6.9 the low resolution approximations (reduced 1 : 4) of the image from Figure 6.6-top, as a function of p. Here, the basis functions are cubic B-splines and the images are interpolated back to the original size. Observe the overshooting (ringing) for high p's in Figure 6.9 (a) and (b). It appears around the nucleus and border of the cells. On the other hand, in Figure 6.9 (c) and (d) the images are much less textured. Subjectively, these approximations are more pleasant visually because the regions are more nearly homogeneous.



(a)



(b)

Figure 6.7: Illustrating the reduction of ringing for ℓ_p-approximations with p close to 1. Basis functions: (a) Linear B-splines. (b) Cubic B-splines.

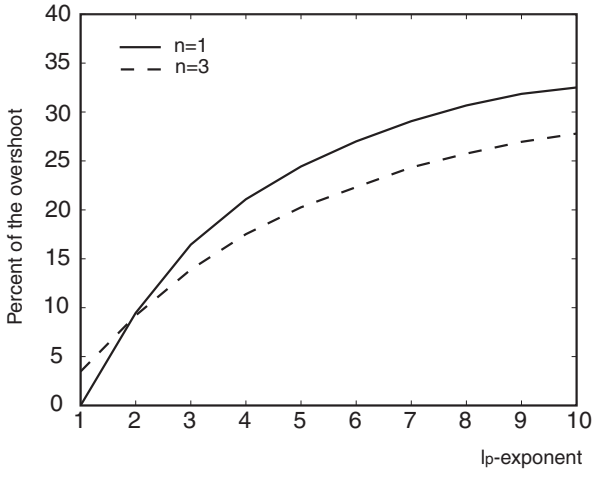


Figure 6.8: Percent of the overshoot as a function of p for linear and cubic spline approximations.

Histogram sparsity

Now, we center our attention on the study of the histograms corresponding to the detail images.

Ideally, we would like our error image to be as sparse as possible, with an histogram presenting a high peak at zero. This would indicate that a large portion of the image is reproduced in the low-resolution approximation. With this idea in mind, we compare in Figures 6.10, and 6.11 the histograms of the detail images for different values of p for a series of biomedical images. In all cases, the sparsest detail histograms correspond to p's close to 1, indicating that the gray value in the original image is more frequently kept in the low resolution approximation than for larger p's. For images in which the amount of noise (due to the characteristics of the image modality) is moderate or low, the height of the peak at zero is impressive.

Furthermore, combining the visual information from the approximated images and the detail histogram, we can derive the following conclusions: The near zero values in the detail histogram correspond to "large" objects in the original

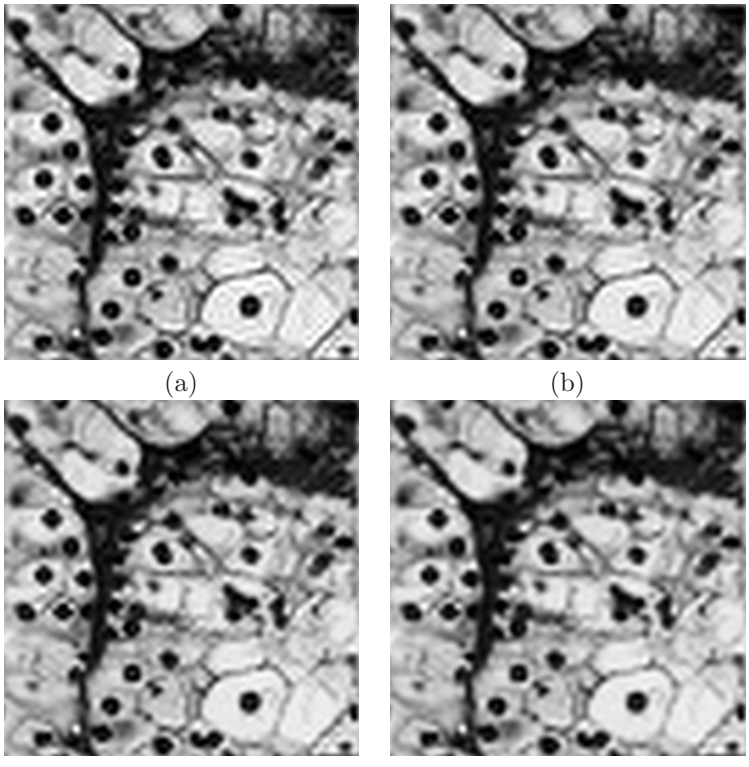


Figure 6.9: Expanded version of the approximations calculated using cubic splines at scale 4 for different values of p. (a) ℓ_3-optimal (b) ℓ_2-optimal (c) $\ell_{1.2}$-optimal (d) $\ell_{1.05}$-optimal

image where the term "large" is relative to the current scale. In other words, the "large" objects and background are kept in the approximation image while "small" objects are retained in the detail images. In addition, we benefit from an excellent preservation of the shape structures for low p's. The images are much more blurred for high p (see Figure 6.9).

The height of the peak at zero and the spread of the detail histogram depend heavily on the characteristics of the image. The peak is higher (resp., lower),

Table 6.3: SNR_{ℓ_p} measures corresponding to the images in Figure 6.9

energy/distance	ℓ_3-optimal	ℓ_2-optimal	$\ell_{1.2}$-optimal	$\ell_{1.05}$-optimal
SNR_{ℓ_3} (dB)	**16.22**	16.08	15.43	15.11
SNR_{ℓ_2} (dB)	17.34	**17.47**	17.21	16.99
$\mathrm{SNR}_{\ell_{1.2}}$ (dB)	18.38	18.79	**18.99**	18.90
$\mathrm{SNR}_{\ell_{1.05}}$(dB)	18.58	19.05	19.37	**19.31**
Kullback-Leibler	0.189	0.163	0.138	**0.130**
Entropy	4.485	4.442	4.397	**4.388**

if there is more (resp., less) edge information in the original image. The averaging effect characteristic of the least-squares approximation leads to a detail histogram with a Gaussian appearance. As p grows, the averaging gets even more accentuated. The spread increases with the degree of uniformity of the original histogram, independently of p.

Quantitative results for the images in Figure 6.9 are given in Table 6.3. Each column correspond to the results coming from a different ℓ_p-approximation.

The table displays the values of the SNR_{ℓ_p}, the Kullback-Leibler distance between the histograms of the original and low resolution images and the entropy of the residues for each of the calculated ℓ_p-approximations.

We observe that the results are consistent: The minimum ℓ_p-error (maximum SNR_{ℓ_p}) is achieved for the corresponding ℓ_p-approximation in each case. The results of the Kullback-Leibler distance clearly indicate that the histogram of the image is best preserved for values of p close to 1. The entropy of the difference images also tends to get smaller for p close to 1. Note that the last results concerning the Kullback-Leibler and entropy measures are nothing more but manifestations of the fact that ℓ_1-approximation tries to preserve the original image values.

Thus, our conclusion is that the most promising scheme is the ℓ_1-approximation because of the following properties:

- Preservation of the structure shapes at different scales which is appropriate for object detection.
- Reduction of ringing and spurious textures.
- The point structures are presented only at the finest scales of the detail images. This may be an advantage for some applications; for example, in the detection of microcalcifications in mammograms (see Figure 6.12).

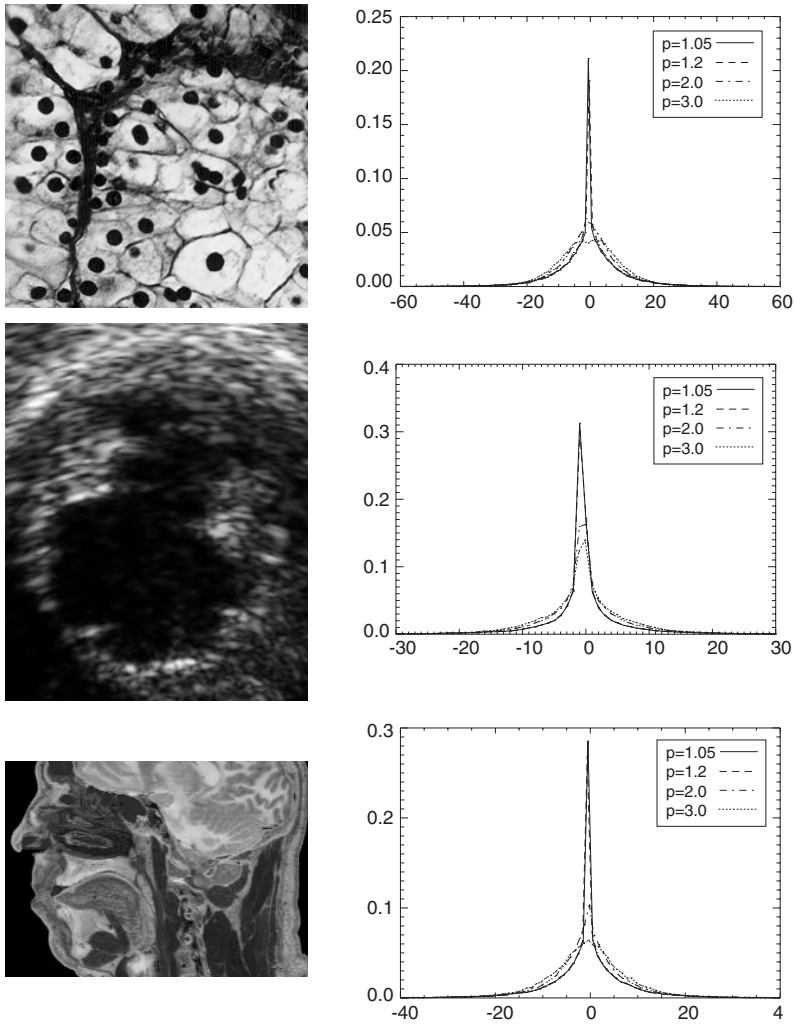


Figure 6.10: (Left) Biomedical images. (Right) Histogram of the detail images (original minus the approximated version at scale 2 using cubic splines as basis functions) for different values of p. Note the high peak at zero for p close to 1.

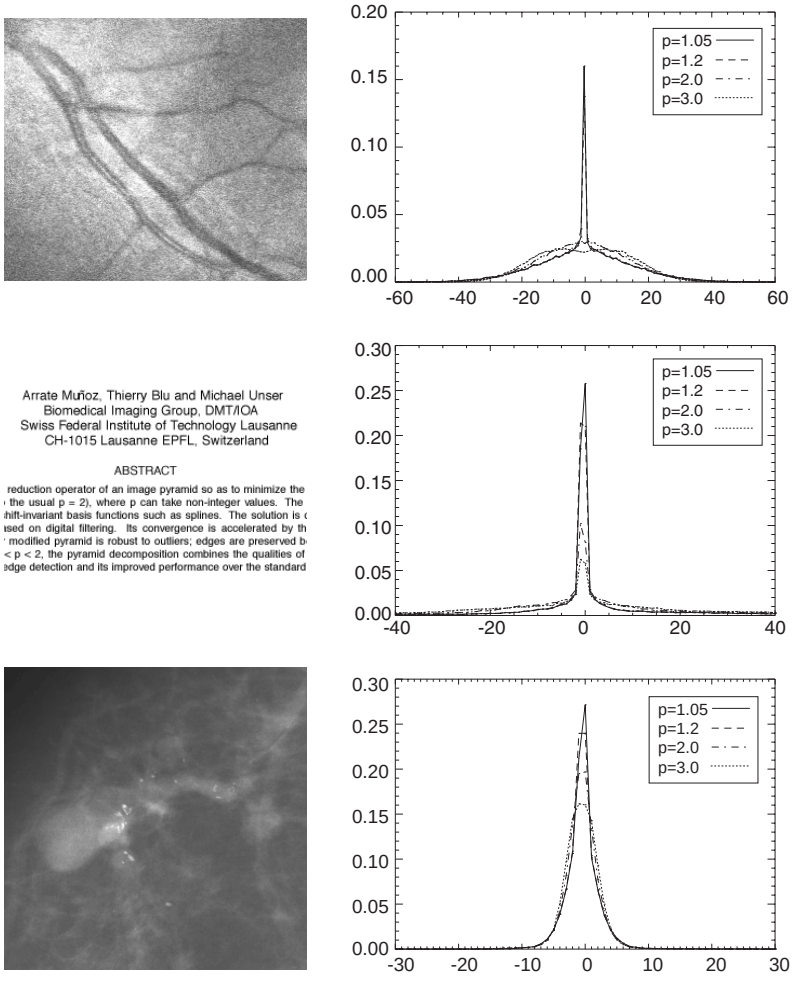
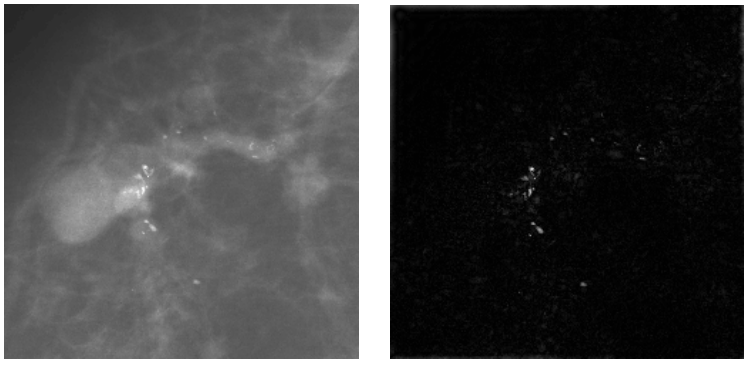


Figure 6.11: (Left) Test images. (Right) Histogram of the detail images (original minus the approximated version at scale 2 using cubic splines as basis functions) for different values of p. Note the high peak at zero for p close to 1.



(a) (b)

Figure 6.12: Illustration of the detection of microcalcifications using ℓ_1-approximations. (a) Mammogram. (b) Difference image for a scale change of 8.

6.3.2 Comparison of approximations for different orders

We now examine the choice of the degree of our spline basis functions. Mainly, we are concerned with the trade-off between quality of approximation and computational complexity. For our experiments we have chosen to compare approximations (reduction $1:4$) calculated using B-splines of degrees 0, 1, 3 and 5 as basis functions. The results are shown in Figure 6.13 and 6.14 for $p = 2$ and $p = 1.05$, respectively. We observe that the blocking artifacts typical of piecewise constant spline approximation (Figure 6.13(a) and 6.14(a)) disappear for higher order splines. Most observers will also agree that the subjective quality of spline approximations with p's close to 1 (Figure 6.14) is better than $p = 2$ (Figure 6.13) for all degrees. Note that for the least-squares case the ringing gets visibly accentuated as the spline degree increases, while this is less the case for $p = 1$. In Table 6.4, we give the SNR_{ℓ_p} and entropy of the difference image that correspond to the ℓ_1 and ℓ_2 approximations calculated for different spline degrees. We observe that we have lower values of the entropy (maximum SNR_{ℓ_p}) with p's close to 1 when using spline basis of the same degree. The minimum ℓ_2-error is reached for the higher order splines. This finding is consistent with the standard theory of splines [22, 127]: as the degree n increases, the spline approximation converges to Shannon's solution which minimizes the L_2-error

Table 6.4: SNR_{ℓ_p} measures and entropy corresponding to the ℓ_1 and ℓ_2-approximations for different scales and degrees of the spline basis functions.

	Degree	$n=0$		$n=1$	
N		ℓ_1	ℓ_2	ℓ_1	ℓ_2
2	SNR_{ℓ_p} (dB)	22.60	19.77	25.74	22.78
	Entropy	3.995	4.043	3.495	3.757
3	SNR_{ℓ_p} (dB)	19.79	16.73	21.39	19.09
	Entropy	4.185	4.383	4.096	4.224
4	SNR_{ℓ_p} (dB)	17.78	15.00	19.01	16.96
	Entropy	4.526	4.610	4.410	4.487
5	SNR_{ℓ_p} (dB)	16.51	13.87	17.62	15.70
	Entropy	4.656	4.767	4.586	4.642
	Degree	$n=3$		$n=5$	
N		ℓ_1	ℓ_2	ℓ_1	ℓ_2
2	SNR_{ℓ_p} (dB)	**26.23**	24.15	25.20	**24.23**
	Entropy	**3.494**	**3.645**	3.648	3.650
3	SNR_{ℓ_p} (dB)	**21.76**	19.87	21.29	**19.92**
	Entropy	**4.082**	**4.157**	4.149	4.159
4	SNR_{ℓ_p} (dB)	**19.31**	17.47	19.02	**17.50**
	Entropy	**4.386**	**4.442**	4.426	4.443
5	SNR_{ℓ_p} (dB)	**17.90**	16.10	17.68	**16.13**
	Entropy	**4.556**	**4.603**	4.583	4.603

when the function is bandlimited or very lowpass (usual case for an image). The main drawback of the least-squares distance measure is that it does not penalize enough oscillations and ringing artifacts. Interestingly, if one looks at the ℓ_1-approximation, the optimal model turns out to be the cubic spline ($n = 3$). It is not surprising the fact that too high an order splines are not good with respect to ℓ_1-approximations because the basis functions tend to $\text{sinc}(x)$ whose samples are not in ℓ_1. This is also consistent with the fact that the ℓ_1-distance is the one that penalizes ringing most.

The ℓ_1-cubic splines are also best in terms of data compression (entropy minimization), combining a good order of approximation with a reduction of artifacts.

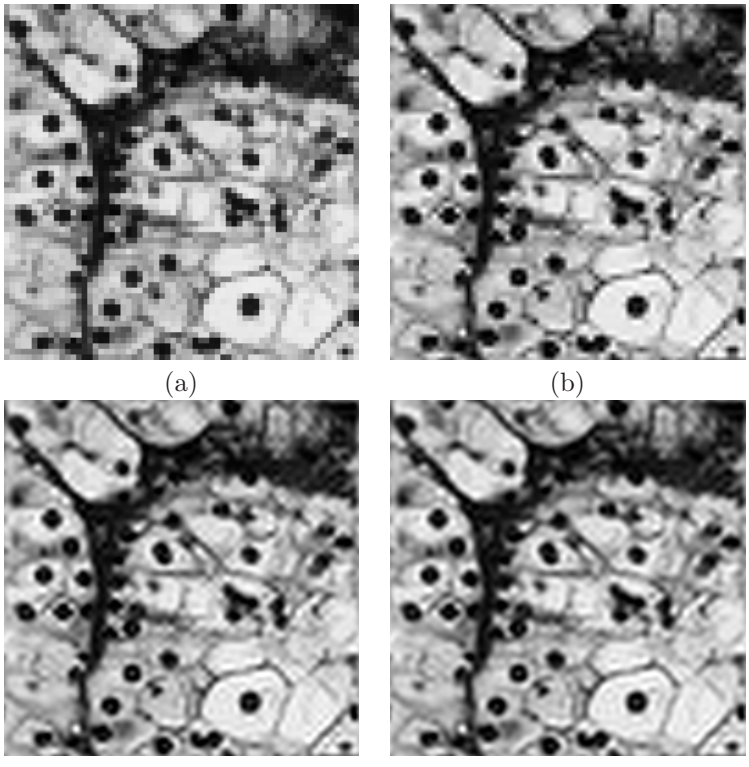
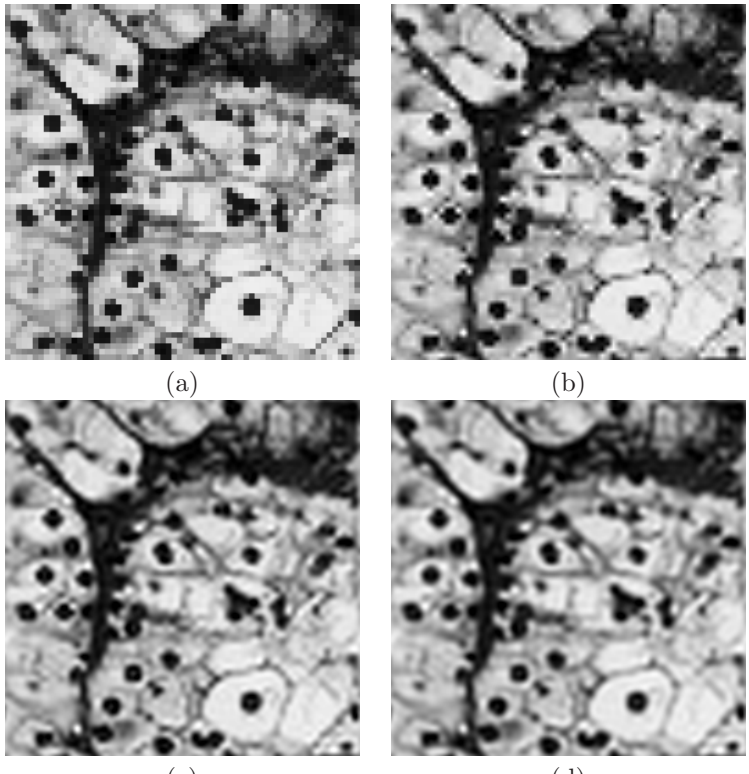


Figure 6.13: Comparison of the least-squares ($p = 2.0$) approximations calculated at scale 4 for different degrees of the splines. (a) $n = 0$ (b) $n = 1$ (c) $n = 3$ (d) $n = 5$. See Table 6.3 for quantitative error information.

6.4 ℓ_p pyramid versus median pyramid

As mentioned in the introduction, median pyramids have been widely used in the literature because of their desirable properties of edge and detail preservation [82, 111].

The reducer operator of the median pyramids computes the decimated version of a median filter output. We can, as with our method, either start from



(a) (b) (c) (d)

Figure 6.14: Comparison of the ℓ_1-approximations calculated at scale 4 for different degrees of the splines. (a) $n = 0$ (b) $n = 1$ (c) $n = 3$ (d) $n = 5$. See Table 6.3 for quantitative error information.

the original image for all the resolution levels, or apply the successive refinement scheme. As we have already pointed out before, the results will be different.

Our model is equivalent to a median pyramid in one particular case: while minimizing the error in the ℓ_1-sense and using as the B-spline of degree 0 as the interpolation function. If the quality of the low-order interpolation is not satisfactory (e.g., because of blocking artifacts), we can simply increase the approximation order by increasing the spline degree. We will still be optimal in

the ℓ_1-sense but our reduction operator will no longer correspond to a median filter. Naturally, the approximation error decreases as our interpolation model improves.

The advantages of our ℓ_p-approximation model over a classical median pyramid can be summarized as follows:

- The reduction operator is consistent with our approximation model.
- The error is minimized in a well-defined sense.
- The existence of an underlying continuous model allows for the evaluation of continuously-defined operators.
- The model is flexible as it is possible to tune the parameter p and the degree of the B-spline which determines the space in which the original image is projected.

6.5 Perceptual relevance of the ℓ_1 metric

There are two related aspects when computing and evaluating image approximations that should be considered:

- The optimization criterion and the algorithm (ℓ_p-projection) used to approximate the input signal.
- The ℓ_p-metric used to measure the approximation error.

Obviously, if we know the metric that best matches our visual perception of image quality, it makes good sense to use the corresponding approximation algorithm.

From a perceptual point of view, what we consider to be a good result depends on the sensitivity of the human observer to details at different frequencies and contrasts [87]. From the examples collected in this thesis and our experimentation with the algorithms, we are tempted to conclude that the ℓ_1-projections look perceptually better than the ones obtained with larger values of p. On the other hand, the error images for ℓ_1-projections also contain details and features that are more noticeable visually. This is consistent with the observation that ℓ_1-approximation has a stronger tendency than others to simplify images.

In [39], DeVore *et al.* performed experiments to determine the L_p-norm that best matched the response of the visual system. They concluded that the L_1-norm was the most appropriate for measuring image compression errors.

They used these results to justify their non-linear wavelet-based compression algorithm. Their findings correlate well with our results. It is clear from the images that we have presented that ringing is disturbing visually. The ℓ_1-norm comes out best because it is the one that penalizes the oscillations most (esp., ringing due to the fact that the sinc is bounded if we measure it with the L_2 norm whereas it is not with L_1).

6.6 Summary

We have presented a theoretical framework for obtaining multiresolution image approximations with non-Euclidean norms. In addition, we have proposed an efficient iterative algorithm based on digital filtering to calculate these approximations. In the experimental part, we found ℓ_1-pyramids to be the most promising ones. Overall, they led to better feature preservation and resulted in less ringing artifacts. They also produced the sparsest error images which is relevant for coding applications. These are all properties that should make them useful for multi-scale processing.

Another interesting finding is that cubic splines gave the best results among all other splines when the approximation was done in the ℓ_1-norm. Unlike the ℓ_2-ranking which always gives the advantage to higher-order approximations, this result correlates well with the fact that the cubic spline model is often the preferred one in applications [125]. Again, this supports the general perception that cubic B-splines offer the best compromise in terms of approximation power versus the support of the basis functions.

Chapter 7

Asymptotic Error Analysis of L_p-approximations

As was discussed in Chapter 2, a critical issue in the approximation theory in a multiresolution context is to measure the decay of the approximation error as the resolution gets finer [112]. In the early 70's, Strang and Fix established the following theorem [113]: Let $\varphi(x)$ be a compactly supported L_2 function that generates the subspace $V_T = \text{span}\{\varphi\left(\frac{x}{T} - k\right)\}_{k\in\mathbb{Z}}$. Then, we have the equivalence

$$\inf_{f_T(x)\in V_T} \|f - f_T\|_{L_2} \quad \leq \quad C \cdot T^L \cdot \|f^{(L)}\|_{L_2} \iff$$

$\varphi(x)$ satisfies the Strang-Fix conditions of order L.

It was generalized to functions $\varphi(x)$ with a suitable fast decay and to L_p-norms by Jia and others [67, 74]. This result implies that higher order approximations converge faster and require less terms (or scales) to approximate a smooth function within a given error tolerance.

A couple of decades later, wavelet researchers got interested in the Strang-Fix theory to characterize the decay of the error of their orthogonal wavelet decompositions [100, 112]. Sweldens showed that the leading constant is an acceptable measure of the efficiency of the wavelet expansions [117, 118]. He also proved that the leading terms of the expansion only depend on the multiresolution subspace and not on how the complementary wavelet subspaces are chosen. In other words, the biorthogonal wavelet expansions are asymptotically optimal

in the sense that the error is the same as in the orthogonal case. Their finding is interesting because biorthogonal transforms leave some freedom in the design of the analysis filters. Unser in [124] extended their analysis by the specification of the optimal constant which led to a sharper and asymptotically exact L_2-estimate. His method is simpler as it avoids the use of wavelet expansions and points out clearly the advantage of splines.

Later on, Blu *et al.* in [21, 22] derived the Fourier domain characterization of the approximation error of L_2-approximations by means of an error kernel, as was presented in Chapter 2. They also found a closed-form expression for the constant in the refinable case. Their research considers all the convolutional approximation operators studied in [67]. Furthermore, the concept of MOMS presented in Chapter 2 was introduced in [20]. This work provides a strong counter example to the commonly-held belief that the smoothness of the basis functions is a strong determinant of the approximation power of the representation.

Closely related to this research is the characterization of the approximation error in the L_p-norm. Wei *et al.* calculated the leading constants of the interpolation and of the L_2-projection of the generalized coiflets (compactly supported orthogonal wavelet) [139]. Dekel *et al.* applies the time-domain method described in [124] to calculate the leading constants of multivariate interpolation using arbitrary basis functions of sufficiently fast decay [38]. In this chapter, we reproduce the derivation of this result for the univariate case and based on it, we deduce our main contributions [1] which are: First, the evaluation of the asymptotic constant for L_p-projections (which are nonlinear and nonconvolutive operators). Second, we shows that it is possible to reach the asymptotic quality (i.e., the asymptotic constant) of the L_p-projection using a convolutive linear operator.

Most of the asymptotic expansions are presented with "$o(\cdot)$" and "$O(\cdot)$" terms. Writing $f(x) = o(x^n)$ is equivalent to writing $\lim\sup_{x\to 0} \left|\frac{f(x)}{x^n}\right| = 0$. Writing $f(x) = O(x^n)$ is equivalent to writing $\lim\sup_{x\to 0} \left|\frac{f(x)}{x^n}\right| < \infty$ (i.e., not necessarily 0).

7.1 Approximation space and methods

We would like to find a reasonable approximation at scale T, $f_T(x)$, of a function $f(x) \in L_p$. Our approximation $f_T(x)$ belongs to the T-integer shift-invariant

[1] joint work with Thierry Blu.

approximation space defined as

$$V_T = \underset{k\in\mathbb{Z}}{\text{span}} \left\{ \varphi\left(\frac{x}{T} - k\right) \right\} \cap L_p, \tag{7.1}$$

where φ is an appropriate generating function.

Thus, the general form of our approximation signal model at resolution T is

$$f_T(x) = \sum_{k\in\mathbb{Z}} c_k(T)\varphi\left(\frac{x}{T} - k\right), \tag{7.2}$$

where the $c_k(T)$'s are the coefficients that describe the input signal $f(x)$ in the approximation space V_T. Their values depend on the scale T.

We say that $\{\varphi\left(\frac{x}{T} - k\right)\}_{k\in\mathbb{Z}}$ forms a p-stable Riesz basis if and only if

$$\forall c \in \ell_p,\ A \cdot \|c\|_{\ell_p} \leq \left\| \sum_{l\in\mathbb{Z}} c_l \varphi\left(x - l\right) \right\|_{L_p} \leq B \cdot \|c\|_{\ell_p} \tag{7.3}$$

with $0 < A, B < \infty$ [3]. This ensures that V_T is a closed subspace of L_p and that each function $f_T(x) \in V_T$ has a unique representation in terms of its coefficients $c_k(T)$.

The concept of order of approximation for $\varphi(x)$ was defined in Chapter 2. An alternative formulation of this property is derived by periodization of the function $x^l\varphi(x)$ whose Fourier transform is $j^l\hat{\varphi}^{(l)}(\omega)$. Using the property that periodization in the time domain corresponds to a sampling in the frequency domain, we obtain the following Fourier series representation:

$$\sum_{k\in\mathbb{Z}} (x-k)^l \varphi(x-k) = j^l \sum_{k\in\mathbb{Z}} \hat{\varphi}^{(l)}(2\pi k) e^{j2\pi kx} = m_l, \quad \text{for } l = 0, ..., L-1, \tag{7.4}$$

where the constant m_l is the lth-order moment of the generating function defined as

$$m_l = \int x^l \varphi(x) dx = j^l \hat{\varphi}^{(l)}(0). \tag{7.5}$$

An Lth-order generating function in the L_p-sense is a function $\varphi(x) \in L_p$ that forms a p-stable Riesz basis and has Lth-order of approximation.

The approximation of $f(x)$ in V_T given by an T-integer shift-invariant linear operator Q_T is

$$Q_T f(x) = \sum_k \gamma_k(T) \varphi\left(\frac{x}{T} - k\right), \tag{7.6}$$

where the coefficients $\gamma_k(T)$ are calculated by the inner-product integral

$$\gamma_k(T) = \int f(\xi) \tilde{\varphi}\left(\frac{\xi}{T} - k\right) \frac{d\xi}{T}. \tag{7.7}$$

As we know already from Chapter 2, the standard interpolation scheme corresponds to $\tilde{\varphi}(x) = \delta(x)$. Then, the approximation coefficients $\gamma_k(T)$ are the samples of the input function $\gamma_k(T) = f(kT)$. Note that the biorthogonality condition $\langle \tilde{\varphi}(x-k), \varphi(x) \rangle = \delta_k$ gives an approximation operator that is a projector. Nevertheless, the biorthogonality of $\varphi(x)$ and $\tilde{\varphi}(x)$, together with having $\tilde{\varphi}(x) \in V_1$ (i.e., $\tilde{\varphi} = \mathring{\varphi}$), does not correspond to the minimum error solution in the L_p-sense.

In the rest of the chapter, we will denote the L_p-projection $P_{T,p} f(x)$ as

$$P_{T,p} f(x) = \sum_{k \in \mathbb{Z}} c_k(T) \varphi\left(\frac{x}{T} - k\right) \tag{7.8}$$

where the coefficients $c_k(T)$ correspond to the minimization of the approximation error in the L_p-sense

$$c_k(T) = \arg\min_c \left(\|f - P_{T,p} f\|_{L_p} \right). \tag{7.9}$$

Consequently, the projection coefficients $c_k(T)$ cannot be calculated by a linear (i.e., prefiltering and sampling) method, but can be obtained thanks to an optimization algorithm of the type given in Chapter 6.

The connection between the present L_p-formulation and the discrete ℓ_p-projection algorithm in Chapter 6 is provided by Riemann's summation formula

$$\forall x_0 \in \mathbb{R}, T \sum_l |f((x_0 + l)T)|^p = \int |f(\xi)|^p \, d\xi + o(1). \tag{7.10}$$

which expresses the asymptotic equivalence between the continuous and discrete p-norms, provided that the underlying function is continuous. Thus, we have

that $\|f-f_T\|_{L_p} = T\|f-f_T\|_{\ell_p}$ as $T \to 0$, which implies that L_p and ℓ_p-projection are equivalents in the asymptotic regime. This suggest that the asymptotic error formulas that we are stablishing here for L_p-projectors are directly transposable to the discrete case as well.

In the next sections, we will characterize the L_p-error for linear and projection approximations in the asymptotic regime, that is, when the sampling step is sufficiently small or when the signal to approximate is sufficiently smooth, as is the case within slowly varying regions of an image. In the process, we will provide sharp estimates for the leading constants that appear in the Strang-Fix theory of approximation.

7.2 Asymptotic L_p-error of linear approximations

In this section, we follow the time-domain method introduced in [124] to characterize the approximation error of biorthogonal operators with respect to the L_p-norm. This characterization will help us to calculate the asymptotic approximation error of the L_p-projections (minimal error solutions).

Let us define

$$e_L(x) = (-1)^L \sum_k (x-k)^L \varphi(x-k). \tag{7.11}$$

We show now that the functions

$$W_n(x) = \sum_k \left(\int (\xi - x)^n \tilde{\varphi}(\xi - k)\, d\xi \right) \varphi(x-k)$$

which will be seen to play a key role in the asymptotic approximation error, take a very simple expression when φ and $\tilde{\varphi}$ are biorthogonal.

Lemma 3 *If φ has an Lth-order of approximation and if $\tilde{\varphi}$ is biorthogonal to φ, then*

$$W_n(x) = \begin{cases} 0 & \text{for } n = 0, ..., L-1 \\ e_L(x) + K & \text{if } n = L, \end{cases}$$

where

$$K = (-1)^{L+1} m_L - j^L \sum_{k \neq 0} \hat{\tilde{\varphi}}(2k\pi) \hat{\varphi}^{(L)}(2k\pi)^*. \tag{7.12}$$

If $\tilde{\varphi}$ satisfies the partition of unity, then $K = (-1)^{L+1} m_L$.

Proof:

We take a polynomial $P(x)$ with $\deg(P) < L$. Consequently, $P(x) \in \text{span}\{\varphi(x-k)\}_{k\in\mathbb{Z}}$. Because of the biorthogonality between $\tilde{\varphi}(x)$ and $\varphi(x)$, we can write

$$P(x) = \sum_k \left(\int P(\xi)\tilde{\varphi}\left(\xi - k\right) d\xi \right) \varphi(x-k).$$

Let $P(x) = Q(x - x_0)$ then

$$Q(x-x_0) = \sum_k \left(\int Q(\xi - x_0)\tilde{\varphi}\left(\xi - k\right) d\xi \right) \varphi(x-k).$$

Let $x = x_0$, then

$$Q(0) = \sum_k \left(\int Q(\xi - x)\tilde{\varphi}(\xi - k) d\xi \right) \varphi(x-k).$$

The first part of Proposition 3 simply follows by considering $Q(x) = x^n$ with $Q(0) = 0$.

For $l = L$, we write $\xi - x = (\xi - k) + (k - x)$ and use the binomial theorem to expand $W_L(x)$ as

$$W_L(x) = \sum_k \sum_{l=0}^{L} \binom{L}{l} \underbrace{\left(\int (\xi-k)^l \tilde{\varphi}\left(\xi - k\right) d\xi \right)}_{=\int \xi^l \tilde{\varphi}(\xi) d\xi = \tilde{m}_l} (k-x)^{L-l} \varphi\left(x-k\right).$$

By exchanging the sums over l and k, we have

$$W_L(x) = \sum_{l=0}^{L} \binom{L}{l} (-1)^{L-l} \tilde{m}_l \left(\sum_k (x-k)^{L-l} \varphi(x-k) \right).$$

Moreover, from equation (7.4), we know that $\sum_k (x-k)^{L-l}\varphi(x-k) = m_{L-l}$ for $L - l = 0, ..., L-1$. We decompose the sum over l into two terms (from 0 to $L-1$ and L), to have

$$W_L(x) = e_L(x) + K$$

as $\tilde{m}_0 = 1$ if the integral of $\tilde{\varphi}(x)$ is one and where $K = \sum_{l=1}^{L} \binom{L}{l} (-1)^{L-l} m_{L-l} \tilde{m}_l$.

To obtain equation (7.12), we take the Lth derivative at $\omega = 0$ of $\left(\hat{\tilde{\varphi}}(\omega)\hat{\varphi}(\omega)^*\right)^{(L)}\Big|_{\omega=0}$ which yields

$$\left(\hat{\tilde{\varphi}}(\omega)\hat{\varphi}(\omega)^*\right)^{(L)}\Big|_{\omega=0} = \sum_{l=0}^{L}\binom{L}{l}\hat{\tilde{\varphi}}^{(l)}(0)\hat{\varphi}^{(L-l)}(0)^*.$$

We rewrite this identity in terms of moments because $\hat{\tilde{\varphi}}^{(l)}(0) = (-j)^l\tilde{m}_l$ and $\hat{\varphi}^{(L-l)}(0) = (-j)^{L-l}m_{L-l}$ and replace the expression that defines K to have

$$\begin{aligned}\left(\hat{\tilde{\varphi}}(\omega)\hat{\varphi}(\omega)^*\right)^{(L)}\Big|_{\omega=0} &= \sum_{l=0}^{L}\binom{L}{l}(-1)^{L-l}(-j)^L m_{L-l}\tilde{m}_l \\ &= (-j)^L\left(K + (-1)^L m_L\right).\end{aligned}$$

Thus, we can express K as

$$K = (-1)^{L+1}m_L + j^L\left(\hat{\tilde{\varphi}}(\omega)\hat{\varphi}(\omega)^*\right)^{(L)}\Big|_{\omega=0}.$$

Finally, we express the biorthogonality condition in the Fourier domain

$$\langle\varphi(x-k),\tilde{\varphi}(x-l)\rangle = \delta_{k,l} \Longleftrightarrow \sum_k \hat{\tilde{\varphi}}(\omega+2k\pi)\hat{\varphi}(\omega+2k\pi)^* = 1$$

which can be rewritten as

$$\hat{\tilde{\varphi}}(\omega)\hat{\varphi}(\omega)^* = 1 - \sum_{k\neq 0}\hat{\tilde{\varphi}}(2k\pi)\hat{\varphi}(\omega+2k\pi)^* + o(\omega^L)$$

whence,

$$\left(\hat{\tilde{\varphi}}(\omega)\hat{\varphi}(\omega)^*\right)^{(L)}\Big|_{\omega=0} = -\sum_{k\neq 0}\hat{\tilde{\varphi}}(2k\pi)\hat{\varphi}^{(L)}(2k\pi)^*$$

and by substitution we have (7.12).

If $\tilde{\varphi}$ satisfies the partition of unity then $\hat{\tilde{\varphi}}(2k\pi) = 0$, for $k \neq 0$ yielding the desired result $K = (-1)^{L+1}m_L$. An example of function $\tilde{\varphi}$ for which K has this value is the dual $\mathring{\varphi}$. So, we denote $\mathring{K} = (-1)^{L+1}m_L$. If the sampling function $\tilde{\varphi}$ is the interpolation prefilter of φ, $\varphi_{\text{int}} = \sum_k p_k\delta(x-k)$ then $\hat{\tilde{\varphi}}(2k\pi) = \hat{\tilde{\varphi}}(0) = 1$. So, the expression for K simplifies to

$$K_{\text{int}} = \mathring{K} + \delta K_{\text{int}} \tag{7.13}$$

where $\delta K_{\text{int}} = -j^L\sum_{k\neq 0}\hat{\varphi}^{(L)}(2k\pi)^*$. ■

Note that $(-1)^L W_L(x)$ is the error for the lowest degree monomial that cannot be approximated exactly by the linear shift-invariant approximation operator $Q_T f(x)$. The lower degree errors $W_n(x)$ are zero as the monomials of degree smaller than L are reproduced exactly.

We now use Lemma 3 to derive an estimate of the approximation error $f(x) - Q_T f(x)$ when $\varphi(x)$ and $\tilde{\varphi}(x)$ are biorthogonal.

Theorem 4 *If $\varphi(x)$ has an Lth-order of approximation, if $f(x) \in C^{L+1}$ and if $\tilde{\varphi}(x)$ and $\varphi(x)$ are biorthogonal, then*

$$f(x) - Q_T f(x) = -\frac{T^L}{L!} f^{(L)}(x) \left(e_L\left(\frac{x}{T}\right) + K \right) + o(T^L). \tag{7.14}$$

and

$$\|f - Q_T f\|_{L_p} = C_{\varphi,p} T^L \|f^{(L)}\|_{L_p} + o(T^L) \tag{7.15}$$

where $C_{\varphi,p} = \dfrac{1}{L!} \left(\displaystyle\int_0^1 |e_L(x) + K|^p dx \right)^{\frac{1}{p}}$.

Proof: Thanks to the partition of unity

$$f(x) = f(x) \sum_k \varphi\left(\frac{x}{T} - k\right),$$

we write

$$f(x) - Q_T f(x) = \sum_k (f(x) - \gamma_k) \varphi\left(\frac{x}{T} - k\right). \tag{7.16}$$

We replace $f(\xi)$ by its Lth-order Taylor series around x

$$\begin{aligned} f(x) - \gamma_k &= \int (f(x) - f(\xi)) \tilde{\varphi}\left(\frac{\xi}{T} - k\right) \frac{d\xi}{T} \qquad (7.17) \\ &= -\int \left(\sum_{n=1}^{L} f^{(n)}(x) \frac{(\xi - x)^n}{n!} + R_{L+1}(\xi) \right) \tilde{\varphi}\left(\frac{\xi}{T} - k\right) \frac{d\xi}{T} \end{aligned}$$

where the remainder of the Taylor series is

$$R_{L+1}(\xi) = \frac{(\xi - x)^{L+1}}{(L+1)!} f^{(L+1)}(\mu) \tag{7.18}$$

with $x \leq \mu \leq \xi$. Next, we rewrite equation (7.16) using the definition of $W_n(x)$ as

$$\begin{aligned} f(x) - Q_T f(x) = & - \sum_{n=1}^{L} \frac{T^n f^{(n)}(x)}{n!} W_n\left(\frac{x}{T}\right) \\ & - \int R_{L+1}(\xi) \sum_k \tilde{\varphi}\left(\frac{\xi}{T} - k\right) \varphi\left(\frac{x}{T} - k\right) \frac{d\xi}{T}. \end{aligned} \quad (7.19)$$

The first $(L-1)$ error terms are zero (polynomial cancellation) as a consequence of Proposition 3. This leads to the following estimate:

$$f(x) - Q_T f(x) = -\frac{T^L}{L!} f^{(L)}(x) \left(e_L\left(\frac{x}{T}\right) + K\right) + o(T^L). \quad (7.20)$$

The asymptotic L_p-error of linear approximations $\|f(x) - Q_T f(x)\|_{L_p}$ can now be expressed as

$$\|f - Q_T f\|_{L_p} = \frac{T^L}{L!} \left(\sum_l \int_{lT}^{lT+T} |f^{(L)}|^p \left| e_L\left(\frac{x}{T}\right) + K \right|^p dx \right)^{\frac{1}{p}} + o(T^L) \quad (7.21)$$

Using the fact that the function $e_L\left(\frac{x}{T}\right)$ in (7.14) is T-periodic, and making the change of variable $x \to (x+l)T$, we write

$$\|f - Q_T f\|_{L_p} = \frac{T^L}{L!} \left(T \sum_l \int_0^1 |f^{(L)}((x+l)T)|^p |e_L(x) + K|^p dx \right)^{\frac{1}{p}} + o(T^L) \quad (7.22)$$

Finally, thanks to Riemann's summation formula (7.10), we obtain

$$\|f - Q_T f\|_{L_p} = C_{\varphi,p} T^L \|f^{(L)}\|_{L_p} + o(T^L) \quad (7.23)$$

where $C_{\varphi,p} = \frac{1}{L!} \left(\int_0^1 |e_L(x) + K|^p dx \right)^{\frac{1}{p}}$.

■

7.3 Asymptotic error of L_p-projection approximations

To characterize the asymptotic error for L_p-projection approximations, we start by expressing mathematically the difference between the linear approximation and the projection coefficients. We choose to express the approximation error as a function of both the linear shift-invariant approximation $P_{T,2}f(x)$ and of the L_p-projection operator $P_{T,p}f(x)$, where the basis function φ is of order L and the sampling function $\tilde{\varphi}$ is its dual $\mathring{\varphi}$

$$f(x) - P_{T,p}f(x) = f(x) - P_{T,2}f(x) + \sum_k (\gamma_k(T) - c_k(T))\varphi\left(\frac{x}{T} - k\right). \quad (7.24)$$

where the $c_k(T)$'s are the coefficients of $P_{T,p}f$ and the $\gamma_k(T)$'s the coefficients of $P_{T,2}f$. As $\varphi(x)$ is a generating function of order L, the approximation order of both the linear approximation $P_{T,2}f(x)$ and of the L_p-projection $P_{T,p}f(x)$ coincide and are of order L [74]. We can therefore write,

$$\begin{aligned} \|f - P_{T,p}f\|_{L_p} &= \left\| f(x) - P_{T,2}f(x) + \sum_k (\gamma_k(T) - c_k(T))\varphi\left(\frac{x}{T} - k\right) \right\|_{L_p} \\ &= O(T^L) \end{aligned} \quad (7.25)$$

and

$$\|f - P_{T,2}f\|_{L_p} = O(T^L).$$

By using the norm inequality $\|f_1 + f_2\| \geq \|f_1\| - \|f_2\|$, we have

$$\|f - P_{T,p}f\|_{L_p} \geq \left\| \sum_k (\gamma_k(T) - c_k(T))\varphi\left(\frac{x}{T} - k\right) \right\|_{L_p} - \|f - P_{T,2}f\|_{L_p}.$$

It follows that the middle term has the same convergence rate

$$\left\| \sum_k (\gamma_k(T) - c_k(T))\varphi\left(\frac{x}{T} - k\right) \right\|_{L_p} = O(T^L). \quad (7.26)$$

Moreover, the lowerbound inequality of the p-stable Riesz basis definition gives

$$A \cdot \|\gamma - c\|_{\ell_p} T^{\frac{1}{p}} \leq \left\| \sum_k (\gamma_k(T) - c_k(T)) \varphi\left(\frac{x}{T} - k\right) \right\|_{L_p} \leq C \cdot T^L. \tag{7.27}$$

This implies that $\gamma_k(T) - c_k(T) \leq \frac{C}{A} \cdot T^{L-\frac{1}{p}} \Rightarrow \gamma_k(T) - c_k(T) = o(T^{L-1})$ (for $p > 1$). Furthermore, if $c_k(T)$ is differentiable L-times with respect to T, then $\gamma_k(T) - c_k(T) = O(T^L)$. Thus, we can conclude that the quantity

$$u_k(T) = \frac{L!}{T^L} \frac{1}{f^{(L)}(kT)} (c_k(T) - \gamma_k(T)); \tag{7.28}$$

remains bounded when $T \to 0$. The sequence $u_k(T)$ is the difference of order L between the coefficients $\gamma_k(T)$ and $c_k(T)$ which coincide in value up to this order.

Next, we prove that the series $u_k(T)$ in (7.28) behave as a constant $u_k(T) \to u$ in the asymptotic regime which simplifies the characterization of the asymptotic error.

To this end, we use the approximation

$$\left| \varphi\left(\frac{x}{T} - k\right) \left(f^{(L)}(x) - f^{(L)}(kT) \right) \right| \leq C \cdot T \cdot \left| \varphi\left(\frac{x}{T} - k\right) \right| \tag{7.29}$$

which is valid provided that $\varphi(x)$ has sufficient decay and $f^{(L)}(x)$ is continuous. Thanks to the result given by Theorem 4 and the above approximation formula, we can express $f(x) - P_{T,p} f(x)$ as

$$\begin{aligned} f(x) - P_{T,p} f(x) &= -\frac{T^L}{L!} f^{(L)}(x) \left(e_L\left(\frac{x}{T}\right) + \mathring{K} + \sum_k u_k(T) \varphi\left(\frac{x}{T} - k\right) \right) \\ &\quad + o(T^L). \end{aligned} \tag{7.30}$$

We substitute this expression in the formula for $\|f - P_{T,p} f\|_{L_p}$ and express the integral as a sum of partial integrals to get

$$\begin{aligned}\|f - P_{T,p}f\|_{L_p} &= \frac{T^L}{L!}\left(\sum_l \int_{lT}^{lT+T} \left|f^{(L)}(x)\right|^p \right. \\ &\cdot \left. \left|e_L\left(\frac{x}{T}\right) + \mathring{K} + \sum_k u_k(T)\varphi\left(\frac{x}{T} - k\right)\right|^p dx\right)^{\frac{1}{p}} + o(T^L).\end{aligned} \tag{7.31}$$

We simplify this result by making the change of variable $x \to (x+l)T$ and by using the fact that the function $e_L(x)$ is 1-periodic. Thus, $e_L\left(\frac{x}{T}\right) = e_L(l+x') = e_L(x')$. The resulting formula is

$$\begin{aligned}\|f - P_{T,p}f\|_{L_p} &= \frac{T^L}{L!}\left(T\sum_l \int_0^1 \left|f^{(L)}((l+x)T)\right|^p \right. \\ &\cdot \left. \left|e_L(x) + \mathring{K} + \sum_k u_{k+l}\varphi(x-k)\right|^p dx\right)^{\frac{1}{p}} + o(T^L).\end{aligned} \tag{7.32}$$

Due to the convexity of $|z|^p$ for $p \geq 1$, the inequality $\left|\sum_l \lambda_l z_l\right|^p \leq \sum_l \lambda_l |z_l|^p$ where $\sum_n \lambda_n = 1$ and $\lambda_n \geq 0\ \forall n$, is satisfied. We apply this result to equation (7.32) by identifying

$$z_l = e_L(x) + \mathring{K} + \sum_k u_{k+l}\varphi(x-k)$$

and

$$\lambda_l = \frac{|f^{(L)}((x+l)T)|^p}{\sum_{l'} |f^{(L)}((x+l')T)|^p}.$$

We obtain the expression

$$\begin{aligned}\|f - P_{T,p}f\|_{L_p} &\geq \frac{T^L}{L!}\left(T\int_0^1 \sum_{l'} \left|f^{(L)}((x+l')T)\right|^p \right. \\ &\cdot \left. \left|e_L(x) + \mathring{K} + \sum_l \sum_k \lambda_l u_{k+l}\varphi(x-k)\right|^p dx\right)^{\frac{1}{p}}.\end{aligned}$$

Thanks to Riemann's summation formula, we have

$$\begin{aligned}\|f - P_{T,p}f\|_{L_p} &\geq \frac{T^L}{L!}\|f^{(L)}\|_{L_p} \cdot \\ &\cdot \left(\int_0^1 \left| e_L(x) + \mathring{K} + \sum_l \sum_k \lambda_l u_{k+l}\varphi(x-k)\right|^p dx\right)^{\frac{1}{p}}.\end{aligned} \tag{7.33}$$

which will be further simplified by reduction of the term $\sum_l \sum_k \lambda_l u_{k+l}\varphi(x-k)$ to a constant value independent of x. The following lemma demonstrates that the simplification is feasible.

Lemma 4 *If $\varphi(x)$ is compactly supported (i.e., support $\varphi(x) \in \left(-\frac{L}{2}, \frac{L}{2}\right)$), then*

$$\sum_l \sum_k \lambda_l u_{k+l}\varphi(x-k) = u + o(T)$$

when $T \to 0$, where u is some constant independent of x and T.

Proof: We have

$$\lambda_l u_{k+l}\varphi(x-k) = \frac{\left|f^{(L)}((x+l)T)\right|^p}{\sum_{l'}\left|f^{(L)}((x+l')T)\right|^p} u_{k+l}\varphi(x-k). \tag{7.34}$$

■

We use the approximation $\sup_{|x-k|\leq\frac{L}{2}} |f((x+l)T) - f((k+l)T)| \leq \sup|f'(\xi)|\frac{L}{2}T$ and the change of variable $k + l' \to l'$ to write

$$\lambda_l u_{k+l}\varphi(x-k) = \frac{\left|f^{(L)}((k+l)T)\right|^p}{\underbrace{\sum_{l'}\left|f^{(L)}(l'T)\right|^p}_{=\frac{1}{T}\|f^{(L)}\|_{L_p}^p + o(T)}} u_{k+l}\varphi(x-k) + o(T).$$

We include now the double summation on l and k to have

$$\sum_l \sum_k \lambda_l u_{k+l}\varphi(x-k) = \sum_l \sum_k \frac{T\left|f^{(L)}((k+l)T)\right|^p}{\|f^{(L)}\|_{L_p}^p} u_{k+l}\varphi(x-k) + o(T).$$

We finalize the proof with the change of variable $(k, k+l) \rightarrow (k, l)$ to simplify the expression into

$$\begin{aligned}\sum_l \sum_k \lambda_l u_{k+l} \varphi(x-k) &= \sum_l \sum_k \frac{T\left|f^{(L)}(lT)\right|^p}{\|f^{(L)}\|_{L_p}^p} u_l \varphi(x-k) + o(T) \\ &= \underbrace{\sum_l \frac{T\left|f^{(L)}(lT)\right|^p}{\|f^{(L)}\|_{L_p}^p} u_l}_{\text{independent of x}} + o(T). \end{aligned} \tag{7.35}$$

where we have applied the partition of unity satisfied by $\varphi(x)$. Consequently, we have the following inequality

$$\|f - Q_T f\|_{L_p} \geq \|f - P_{T,p} f\|_{L_p} \geq \frac{T^L}{L!} \|f^{(L)}\|_{L_p} \left(\int_0^1 |e_L(x) + u|^p \, dx \right)^{\frac{1}{p}} \tag{7.36}$$

where u is some constant independent of x.

Theorem 5 *If $\varphi(x)$ has an Lth-order of approximation, if $f(x) \in C^{L+1}$ then,*

$$f(x) - P_{T,p} f(x) = -\frac{T^L}{L!} f^{(L)}(x) \left(e_L\left(\frac{x}{T}\right) + u \right) + o(T^L) \tag{7.37}$$

where $u = \arg\min_u \int_0^1 |e_L(x) + u|^p dx$ and

$$\|f - P_{T,p} f\|_{L_p} = C_{\varphi,p}^{\min} \cdot T^L \cdot \|f^{(L)}(x)\|_{L_p} + o(T^L) \tag{7.38}$$

where $C_{\varphi,p}^{\min} = \frac{1}{L!} \min_u \left(\int_0^1 |e_L(x) + u|^p dx \right)^{\frac{1}{p}}$.

Proof: The result from Theorem 4 and (7.36) implies the following inequalities

$$\begin{aligned}\frac{1}{L!} \left(\int_0^1 |e_L(x) + u|^p dx \right)^{\frac{1}{p}} &\leq \\ &\leq \frac{1}{T^L} \frac{\|f - P_{T,p} f\|_{L_p}}{\|f^{(L)}\|_{L_p}} \leq \\ &\leq \frac{1}{L!} \left(\int_0^1 |e_L(x) + K|^p dx \right)^{\frac{1}{p}} \ \forall K \end{aligned} \tag{7.39}$$

for some u. By definition, the L_p-projector has the smallest error over all approximations so that

$$\frac{1}{T^L}\frac{\|f - P_{T,p}f\|_{L_p}}{\|f^{(L)}\|_{L_p}} \leq \frac{1}{L!}\min_K \left(\int_0^1 |e_L(x) + K|^p dx\right)^{\frac{1}{p}}$$

which implies that the above inequality has to be an equality. The detailed argument rest on the fact that we can construct a linear approximation $Q_T f$ that achieves this upper bound (Theorem 6).

Theorem 4 shows that a linear approximation of the form $Q_T f$ is characterized by a pointwise approximation error given by (7.14) and a L_p-approximation error given by (7.15). The analogy between (7.15) and (7.38) indicates that, if we choose $\tilde{\varphi}$ such as $K = u$, then the linear approximation method reaches the quality of the L_p-projection. This also means that $Q_T f = P_{T,p}f + o(T^L)$. ■

Usually, the value $K = u$ is different from $\mathring{K}$ which is optimized for the L_2-error measure. Thus, the difference $u - \mathring{K}$ is a bias of the L_p-projection with respect to the L_2 (orthogonal) projection.

We characterize the error of an asymptotic approximation for spline based L_p-projections in Figure (7.1). The generating functions are centered B-splines. The value of the $L_p - L_2$ projection bias with respect to the L_p-exponent is shown in Figure 7.1(a) for a spline approximation of order 1, and (b) for an order 4. For odd orders, the value of the bias is null for all p because $e_L(x)$ is anti-symmetric and of zero mean. For even orders, $e_L(x)$ is anti-symmetric with a non-zero mean, so the bias is non-zero except for $p = 2$ as it corresponds to a least-squares projection. This point marks a transition between positive and negative values of the bias. The asymptote is given by the mean of the maximum and minimum values of $e_L(x)$: $\text{Bias}_\infty = -0.0021$. The value of the approximation constant $(C^{\min}_{\beta^{L-1},p})^{\frac{1}{L}}$ with respect to the L_p-exponent is represented in Figure (7.1) (c) for a spline approximation of order 1, and in Figure (7.1) (d) for an order 4. The asymptotes, represented by the dashed lines, correspond to $C_{\beta^0,\infty} = 0.5$ and $(C_{\beta^3,\infty})^{\frac{1}{4}} = 0.19$, respectively.

The next theorem demonstrates that we can construct a linear operator that has the same asymptotic performance as the L_p-projection.

Theorem 6 *Given the basis function φ of approximation order L, we can replace the L_p-projection $P_{T,p}f$ by the linear operator $Q_T f$ in which the sampling function $\tilde{\varphi}$ is defined as*

$$\tilde{\varphi} = (1 - \alpha)\mathring{\varphi} + \alpha\varphi_{\text{int}}$$

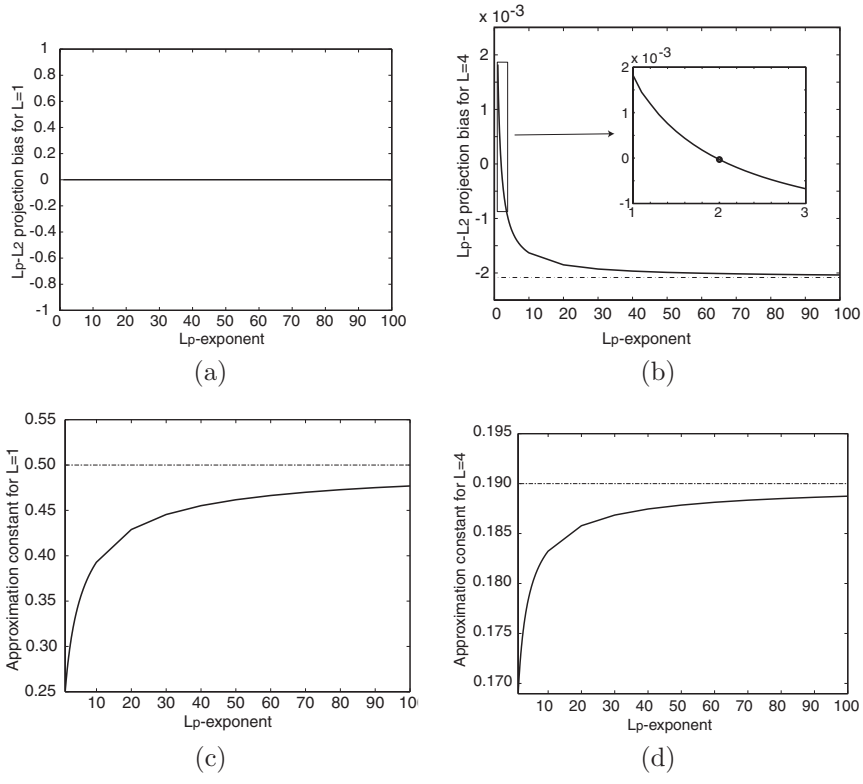


Figure 7.1: Characterization of the error of an asymptotic approximation for spline based L_p-projections. Top: $L_p - L_2$ projection bias $(u - \mathring{K})$, as a function of p. (a) $L = 1$: the bias is zero for all value of p. (b) $L = 4$: the bias is only zero for $p = 2$ (orthogonal L_2-projection). Bottom: Normalized asymptotic approximation constant $(C^{\min}_{\beta^{L-1},p})^{\frac{1}{L}}$ as a function of p. (c) $L = 1$; (d) $L = 4$. The asymptote $p = \infty$ provides the values for L_∞-norm: $\text{Bias}_\infty = -0.0021$, $C_{\beta^0,\infty} = 0.5$ and $(C_{\beta^3,\infty})^{\frac{1}{4}} = 0.19$.

with $\alpha = \frac{u+(-1)^L m_L}{-j^L \sum_{k\neq 0} \hat{\varphi}^{(L)}(2k\pi)^*}$. *With this choice,* $Q_T f$ *and* $P_{T,p} f$ *yield the same asymptotic error up to* $o(T^L)$.

Proof:

$Q_T f$ and $P_{T,p} f$ are both approximations of order L. The magnitudes of the approximation error differ proportionally to the value of the respective approximation constant: $C_{\varphi,p}$ and $C_{\varphi,p}^{\min}$. For the linear approximation $Q_T f$, we choose $\tilde{\varphi} = (1-\alpha)\mathring{\varphi} + \alpha\varphi_{\text{int}}$ which is biorthogonal to φ. Due to linearity, the value of the leading constant $C_{\varphi,p}$ is

$$C_{\varphi,p} = \frac{1}{L!}\left(\int_0^1 |e_L(x) + \mathring{K} + \alpha\delta K_{\text{int}}|^p dx\right)^{\frac{1}{p}}.$$

where $\delta K_{\text{int}} = -j^L \sum_{k\neq 0} \hat{\varphi}^{(L)}(2k\pi)^*$. For the nonlinear approximation $P_T f$ the constant $C_{\varphi,p}^{\min}$ is

$$C_{\varphi,p}^{\min} = \frac{1}{L!}\left(\int_0^1 |e_L(x) + u|^p dx\right)^{\frac{1}{p}}.$$

Thus, we obtain the equivalence (i.e.,$C_{\varphi,p} = C_{\varphi,p}^{\min}$) by making $\alpha = \frac{u-\mathring{K}}{\delta K_{\text{int}}}$. ■

Consequently, we just need to tune the value of α proportionally to the theoretical bias u. Then, we can calculate by filtering the linear approximation coefficients (equation 7.7) that can serve as a good initialization of the L_p-projection minimization algorithm.

Since we work with B-splines as basis functions, we give the specialized form of Theorem 6 for this case.

Corollary 1 *Given the B-spline* β^{L-1} *of approximation order* L*, we can replace the* L_p*-projection* $P_{T,p} f$ *by the linear operator* $Q_T f$ *in which the sampling function* $\tilde{\varphi}$ *is defined as*

$$\tilde{\varphi} = (1-\alpha)\mathring{\beta}^{L-1} + \alpha\beta_{\text{int}}^{L-1}$$

with

$$\alpha = \begin{cases} 0 & \text{for odd } L \\ \frac{u+(-1)^L m_L}{(-1)^{\frac{L}{2}+1} 2\zeta(L)\frac{L!}{(2\pi)^L}} & \text{for even } L, \end{cases}$$

with this choice, $Q_T f$ *and* $P_{T,p} f$ *yield the same asymptotic error up to* $o(T^L)$.

Proof: What are need to do is to evaluate the value of α in Theorem 6 when $\varphi(x) = \beta^{L-1}(x)$. For this purpose, we calculate

$$\delta K_{\text{int}} = -j^L \sum_{k\neq 0} \hat{\varphi}^{(L)}(2k\pi)^*.$$

The Fourier transform of a B-spline of order L (β^{L-1}) is

$$\hat{\varphi}(\omega) = \left(\frac{1-e^{-j\omega}}{j\omega}\right)^L,$$

and thus

$$\begin{aligned}\sum_{k\neq 0} \hat{\varphi}(2k\pi)^* &= \left((1-e^{-j\omega})^L \sum_{k\neq 0} \frac{1}{(j(\omega+2k\pi))^L}\right)^* \\ &= \left(\left((j\omega)^L + o(\omega^L)\right)\left(\sum_{k\neq 0}\frac{1}{(j2k\pi)^L} + o(1)\right)\right)^*. \quad (7.40)\end{aligned}$$

Using the Riemann zeta function, we have

$$\sum_{k\neq 0}\frac{1}{(j2k\pi)^L} = \begin{cases} 0 & \text{for odd L} \\ (-1)^{\frac{L}{2}} 2\zeta(L)\frac{1}{(2\pi)^L} & \text{for even L.}\end{cases}$$

Finally,

$$\sum_{k\neq 0} \hat{\varphi}^{(L)}(2k\pi)^* = \begin{cases} 0 & \text{for odd L} \\ 2\zeta(L)\frac{L!}{(2\pi)^L} & \text{for even L.}\end{cases}$$

i.e., δK_{int} for spline approximation is

$$\delta K_{\text{int}} = \begin{cases} 0 & \text{for odd L} \\ (-1)^{\frac{L}{2}+1} 2\zeta(L)\frac{L!}{(2\pi)^L} & \text{for even L.}\end{cases}$$

■

In this way, we have all the necessary tools to calculate the value of α and then, estimate the value of the L_p-projection coefficients for a B-spline interpolation model. The good news is that we can simply use the L_2-projection as initialization when the degree of the spline is even; otherwise, a little correction needs to be done as indicate above.

7.4 Conclusion

The main contribution of the chapter is the characterization of the error of an asymptotic approximation for L_p-projections. The difficulty comes from the nonlinear structure of the corresponding approximation operator. To derive a solution, we have used the fact that linear projection operators give the same order of approximation as L_p-projection ones when the approximation space is the same. We conclude that we can initialize the L_p-projection algorithm with a well-specified linear solution as the estimates converge to each other in the smooth regions.

Chapter 8

Conclusion

8.1 Novel multiresolution image approximations

In this thesis, we developed two types of extensions to multiresolution image approximations: image resizing for arbitrary scales, and nonlinear methods based on the calculation of ℓ_p-projections.

In the first part (Chapter 4), we proposed an optimal spline-based algorithm for the enlargement or reduction of digital images with arbitrary (noninteger) scaling factors. This projection-based approach can be computed exactly thanks to a new finite-difference method that allows the computation of inner products with analysis functions that are B-splines of any degree n. A noteworthy property of the algorithm is that the computational complexity per pixel does not depend on the scaling factor a. For a given choice of basis functions, the results of our method are consistently better than those of the standard interpolation and resampling procedure; the present scheme achieves a reduction of artifacts such as aliasing and blocking and a significant improvement of the signal-to-noise ratio. The finite-difference method can be applied to other image processing related problems where computations of inner products or of convolutions with splines of different widths are necessary. Other examples described in this thesis are nonuniform to uniform resampling, and the computation of the continuous wavelet transform.

Our second key contribution was to propose an alternative to linear multiresolution analysis. Our approach differs from the morphological or rank-order multiresolution decompositions that have been investigated by other authors.

The underlying continuous-discrete model used in our formalism is the same as the wavelet-like methods: a polynomial spline with a knot spacing that is matched to the resolution. The choice of the spline model specifies the corresponding expansion mechanism which is equivalent to a polynomial spline interpolator. Our departure from the traditional wavelet-like formulation is that the reduction operator is chosen to minimize the approximation error in the ℓ_p-sense (not restricted to the usual $p = 2$), where p can take non-integer values. This leads to a Banach space formulation of the problem which calls for a non-linear solution. Our algorithm uses the same linear components (filters and down/upsamplers) but finds the solution iteratively. While we did not have enough time to pursue applications, we believe that our non-linear pyramids should be useful for most image processing tasks for which multiresolution makes sense, in particular, those that call for a continuous formulation of the problem. In some sense, we are combining the advantages of wavelet-based methods (the availability of a signal model) and morphological ones which are more robust and less prone to ringing artifacts.

8.2 Future research

8.2.1 Applications of ℓ_p-projections with $1 \leq p \leq 2$

Denoising

Thresholding of the transform coefficients of linear multiresolution methods is a state-of-the-art technique for denoising [40]. Some research in this direction shows that this method does not perform well with non-Gaussian (heavy tailed) noise. This reason motivates the research on denoising using nonlinear multiresolution approximations that might be more appropriate to treat non-Gaussian noise [42]. The first step should be an analysis of the statistical properties of the transform coefficients of our spline pyramids, which is necessary to make an optimal choice for the threshold values. Furthermore, one could implement adaptive thresholding rules (such as context modeling [30] or hidden Markov models [33]), which has been shown to improve results in the linear case.

Feature sieve

Nonlinear filters are frequently used to perform a size sensitive decomposition. A nonlinear filter removes features smaller than its associated structuring element, while leaving the larger features practically untouched. A pyramidal adaptation

of the approach can be useful in pattern recognition [53]. In our pyramidal model, the selection of the spline model and of the scale change factor for the reduction step will determine the size of the features extracted in the difference images.

Multiscale algorithms

Multiscale algorithms based on morphological pyramids allow for both a considerable tolerance to non-Gaussian noise and a saving in computational time. Edge detection [138], image registration [61], and video tracking [108] are examples of applications where morphological pyramids significantly enhance performance over comparable linear methods.

In our case, we may take advantage of the underlying continuous model to evaluate image gradients and differentials at various scales, in a way that is globally consistent with our assumptions and approximation mechanism.

8.2.2 Extensions to other measures

In the field of wavelets, several authors have developed algorithms for optimal basis selection by minimizing diversity measures [41, 140]. Basis selection has applications in linear inverse problems where the solution is known or required to be sparse [50, 66]. The use of the term 'diversity' refers here to a measure of antisparsity. Note that minimizing diversity (antisparsity) is equivalent to maximizing concentration (sparsity).

A popular diversity measure is $E^p(x)$, where

$$E^p(x) = \sum_k |x_k|^p, 0 \leq p \leq 1.$$

The motivation for these diversity measures is that their minimization will result in sparse solutions for the detail images. It is well known that, for $p < 1$, ℓ_p is not a true norm. The diversity measure for $p = 0$ is a direct measure of sparsity that provides a count of the number of nonzero elements of a vector x

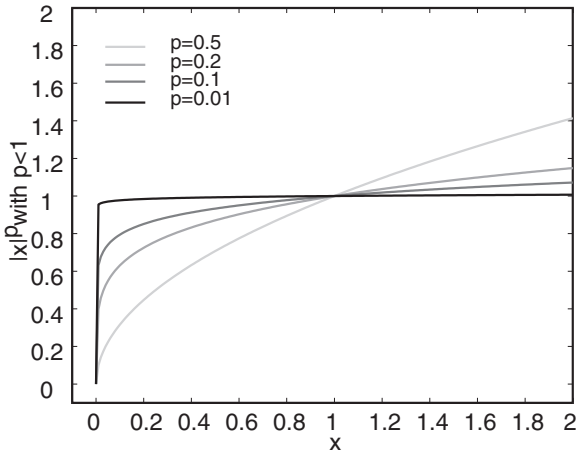
$$E^0(x) = \sharp\{k : x_k \neq 0\}.$$

Finding a minimum for this measure requires an enumerative search and is NP complete. Conversely, the $E^p(x)$ measures for $p < 1, p \neq 0$ are somewhat more amenable to optimization techniques, even though their associated cost functions are concave. The minimization of concave functions is known to be

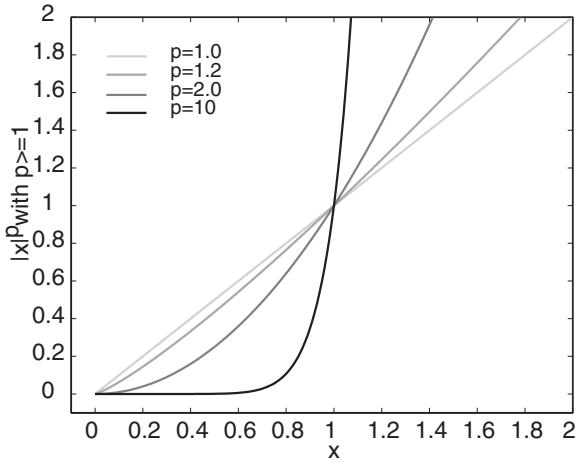
difficult because of the existence of multiple minima. Nevertheless, minimization algorithms with provable convergence can be found in the literature [96].

Figure 8.1 illustrates the tendency of $E^p(x)$ to favor sparse solutions, with a large quantity of null errors and a few 'whatever' ones that contribute modestly. The behavior of $\sum_k |x_k|^p$ for $p >> 1$ is in some sense the opposite; the solutions with a large quantity of small but non-null errors are to be preferred.

Consequently, we expect that designing reduction operators that minimize this type of diversity measures $E^p(x)$ would increase the sparsity over the present solutions.



(a)



(b)

Figure 8.1: $|x_k|^p$ versus x. (a) For $p < 1$: Outliers have a small effect. Solutions having a large quantity of null errors are preferred. (b) For $p \geq 1$: Outliers have a strong effect. Solutions having a large quantity of non-null errors are preferred.

Appendix A

Continuous Wavelet Transform with Arbitrary Scales

The continuous wavelet transform (CWT) of a signal f with the wavelet ψ is defined as

$$W_\psi f(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(x)\psi\left(\frac{b-x}{a}\right) dx. \tag{A.1}$$

It can be interpreted as the correlation of the input signal with a time-reversed version of ψ rescaled by a factor of a. For a 1-D input signal, the result is a 2-D description of the signal with respect to time b and scale a. The scale a is inversely proportional to the central frequency of the rescaled wavelet $\psi_a(x) = \psi(\frac{x}{a})$ which is typically a bandpass function; b represents the time location at which we analyze the signal. The larger the scale a, the wider the analyzing function ψ_a, and hence the smaller the corresponding analyzed frequency. The output value is maximized when the frequency of the signal matches that of the corresponding dilated wavelet. The main advantage over the Fourier Transform (FT) analysis is that the frequency description is localized in time. The advantage over the Short-Time Fourier Transform (STFT) is that the window size varies; low frequencies are analyzed over wide time windows, and high frequencies over narrow time windows, which is more effective than

to use a fixed-size analysis. Typical applications of the CWT are the detection and characterization of singularities [16, 85], pattern recognition [35], image processing [28, 93], fractal analysis [9, 79, 136], noise reduction [75] and the analysis of biomedical signals [49, 64, 141].

The main contribution of this chapter is the development of a fast algorithm for the computation of the CWT at any real scale a and integer time localization b. Mallat's fast wavelet algorithm [79] uses the multiresolution properties of the wavelet to compute the CWT at dyadic scales $a = 2^i$ and time shifts $b = 2^i k, k \in Z$ [99]; it achieves an overall $O(N)$ complexity. Other techniques compute the Wavelet Transform at dyadic scales and integer time points with an *'à trous'* approach. Their complexity per scale is $O(N)$ the same as Mallat's algorithm, but with a larger leading constant [16, 29, 58, 97].

Despite their speed, these methods may not be precise enough for some applications, since a dyadic scale progression cannot be finer than an octave subband decomposition. To achieve a better scale resolution, other approaches have been proposed, either based on M-band decomposition inside an octave [99], [137] or on a generalization of the two-scale relation to general integer N-scale relations [57], [132]. However, none of these algorithms can handle arbitrary scales.

Our purpose here is to develop a novel and fast algorithm that works for any real value of a. It takes advantage of a B-spline decomposition of the input signal and of the mother wavelet. The method exploits the fact that B-splines are compactly supported and that the convolution of two B-splines can be expressed analytically as was explained in Chapter 3.

The chapter is organized as follows. In Section A.1, the spline wavelets are presented. In the next section, the CWT algorithm is mathematically derived and its implementation presented. The chapter ends with the comparison of computational time of our method and a FFT-based one, and the analysis of a biomedical signal.

A.1 Spline wavelets

Among all existing wavelet bases, B-spline wavelets have the advantage of possessing an explicit formula [6]; most wavelets are defined only implicitly by means of a refinement filter. For example, the well-known Haar wavelet is a weighted sum of two B-splines of degree 0. Other wavelets, such as the first derivative or the second derivative of a Gaussian (Mexican hat wavelet), can be closely approximated by linear combination of B-splines of sufficiently high degrees ($n \geq 2$) [123].

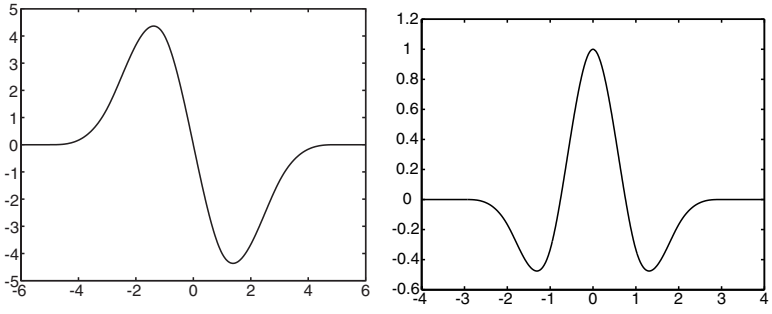


Figure A.1: (a) First derivative of the quartic B-spline enlarged by a factor 2. (b) Second derivative of the quintic spline wavelet.

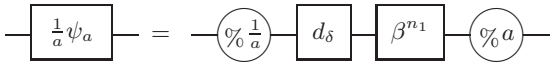


Figure A.2: Schematic representation of a spline wavelet $\frac{1}{a}\psi_a(x)$ using our operator formalism.

The description of wavelets in a B-spline basis allows for an efficient computation of the convolution products of the CWT that takes advantage of the convolution properties of B-splines. Thus, we choose to express our mother wavelet ψ in a B-spline basis of order n_1. The wavelet at scale a is represented by its B-spline expansion

$$\frac{1}{a}\psi_a(x) = \psi\left(\frac{x}{a}\right) = \sum_{k=-K}^{K} d_k \beta^{n_1}\left(\frac{x}{a} - k\right) \tag{A.2}$$

where d_k are the B-spline coefficients (see [129] for additional information on how to choose their value).

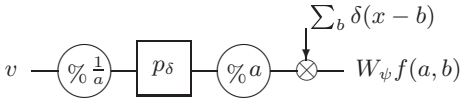


Figure A.3: Graphical representation of our CWT algorithm as a mixed convolution.

A.2 Continuous wavelet transform algorithm

A.2.1 Mathematical derivation

Using the expression for the B-spline inner product (Equation 3.11), the CWT (A.1) for the wavelet (A.2) becomes:

$$\begin{aligned} W_\psi f(a,b) &= \frac{1}{\sqrt{a}} \left(f * \psi_a\right)(b) \\ &= \frac{1}{a^{n_1+\frac{1}{2}}} \left(\Delta_a^{n_1+1} * d_{\delta,a} * D^{-(n_1+1)} f * \delta\left(\cdot + a\frac{n_1+1}{2}\right)\right)(b) \end{aligned}$$

As the intersample distance a is the same for the distributions $\Delta_a^{n_1+1}$ and $d_{\delta,a}$, we can express the continuos wavelet transform as the mixed convolution

$$W_\psi f(a,b) = \left(\left(\sum_{k=-K}^{n_1+1+K} p(k)\delta(\cdot - ak)\right) * v(\cdot)\right)(b), \tag{A.3}$$

where

$$p_k = \frac{1}{a^{n_1+\frac{1}{2}}}(d * q)_k = \frac{1}{a^{n_1+\frac{1}{2}}} \sum_{l=0}^{n_1+1} d(k-l)q(l) \tag{A.4}$$

being q_l the finite differences coefficients in (3.11) and $v(x)$ is given by equation (3.16). The graphical representation is given in Figure A.3.

Spline input signal

If f is a spline that interpolates the discrete input samples s_k, we use the results derived in Chapter 3 and rewrite $v(x)$ as given in equation (3.19).

We introduce this relation in the CWT expression (A.3) which yields,

$$W_\psi f(a,b) = \sum_{k=-K}^{K+n_1+1} \sum_{i=0}^{n_1+n_2+1} g_{i+i_0} w_{a,b}(k,i), \tag{A.5}$$

where $i_o = \left\lceil b - ak + \tau - \frac{n_1+n_2+2}{2} \right\rceil$ and where

$$w_{a,b}(k,i) = p_k \beta^{n_1+n_2+1} \left(b - ak - i - i_0 + \tau \right). \tag{A.6}$$

We have used here the compact-support property of B-splines to reduce the number of terms of the sum over i. In this way, the computation of the CWT reduces to the inner product with the coefficients g_i of a series of precalculated weights $w_{a,b}(k,i)$ (which we can store in a look-up table).

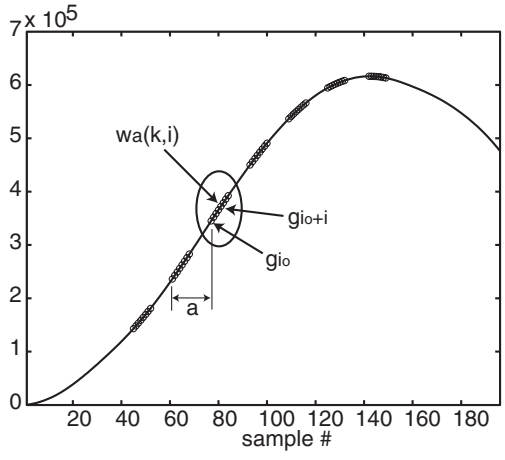


Figure A.4: Spatial structure of the filter w_a: 'clusters' of weights separated by a distance a of each other. Parameters: $n_1 = 3$, $n_2 = 3$, $K = 1$, $a = 16$ and $b = 128$.

In practice, we are typically only interested in the values of b that correspond to the time locations of the original samples, that is, for integer b. Then, we can use the fact that $w_{a,b}(k,i) = w_{a,0}(k, i-b)$, if $b \in \mathbb{Z}$. The algorithm (A.5) is then equivalent to a discrete convolution. This reduces considerably the number of weights to be precalculated, since only the values $w_{a,0} = w_a$ are

required.

Note the interesting spatial structure of the filter w_a (see Figure A.4). The CWT computation consists in filtering the integrated interpolation coefficients g_δ with $(2K + n_1 + 2)$ 'clusters' of length $(n_1 + n_2 + 2)$, each cluster being separated from its neighbors by a distance a. This can be seen as a kind of modified '*à trous*' filter.

A.2.2 Fast implementation

Let us now describe the fast algorithm based on the expansion (A.5). In the initialization step, the B-spline expansion coefficients c_δ of the sampled signal f are calculated, and the running-sum operator Δ^{-1} is applied (n_1+1) times; it is computed recursively by iterating (3.4) (see Section 4.2.4 for the implementation details).

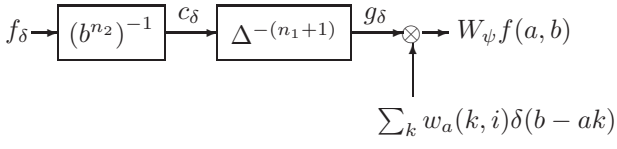


Figure A.5: Schematic representation of the fast wavelet transform. $(b^{n_2})^{-1}$: Computation of the interpolation coefficients c_δ. $\Delta^{-(n_1+1)}$: Calculation of the $(n_1 + 1)$-fold integral of c_δ. $w_a(k, i)$: Look-up table calculation where $k \in [-K, K + n_1 + 1]$, $i \in [0, n_1 + n_2 + 1]$ and $a \in [a_1, a_N]$. N is the number of scales. w_a: Filtering with the mask calculated for each scale. $W_\psi f(a, b)$: CWT of f for scale a at position b.

The intermediate result g_δ does not depend on the scale a. For a given scale a, we compute the weights w_a and store them in a 2-D look-up table of dimensions $(2K+n_1+2)\times(n_1+n_2+2)$. These values are then convolved with the precomputed sequence g_δ. The values w_a and the inter-cluster distance for the filtering depend on a, but the computational complexity is constant and does not depend on a. Moreover, the values w_a do not depend on the signal f. Thus, the computational complexity per point only depends on the values of K, n_1 and n_2. Note that the independence between scales allows for a straightforward parallel implementation (Figure A.5).

A.3 Experimental results

Here, we discuss the implementation of our fast CWT algorithm and compare its execution time with a FFT-based implementation. As example of application, we show the analysis of a biomedical signal.

A.3.1 Comparison with FFT-based computation

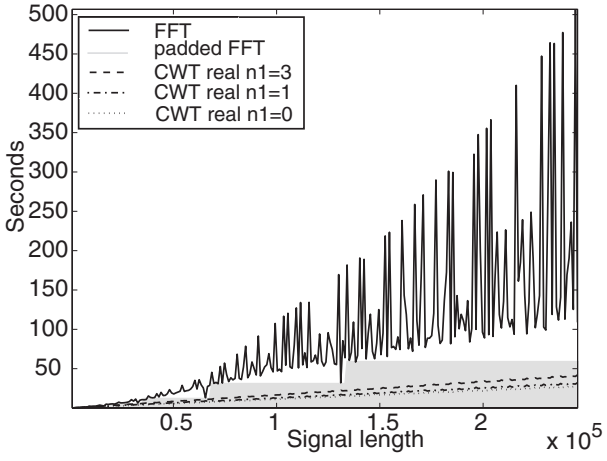


Figure A.6: Comparison of the experimental computation times of the FFT and of our B-spline-based method to calculate the CWT.

The FFT has an overall $O(N \log N)$ complexity and is therefore asymptotically slower than our method which has an $O(N)$ complexity. This can be observed from the experimental comparison of the computation times shown in Figure A.6. We see that, for long input signals, our method is indeed faster. The interpolation degree for the input signal was zero. The CWT was computed over four octaves with 12 scales per octave. The wavelet was the second derivative of the quintic (see Figure A.1 (b)), quartic and cubic B-spline, respectively [132]. The FFT-based method used a radix-2 algorithm when the signal length was a power of 2 and a mixed-radix method for other signal lengths (MATLAB's FFT algorithm). The region where the algorithm is faster than the FFT-radix 2 method with zero padding is colored in gray. The time required to compute the wavelet in the time domain before its FFT computation was neglected. A

parallel implementation for the scale-dependent part of each algorithm would speed up the computations.

A.3.2 Analysis of a biomedical signal

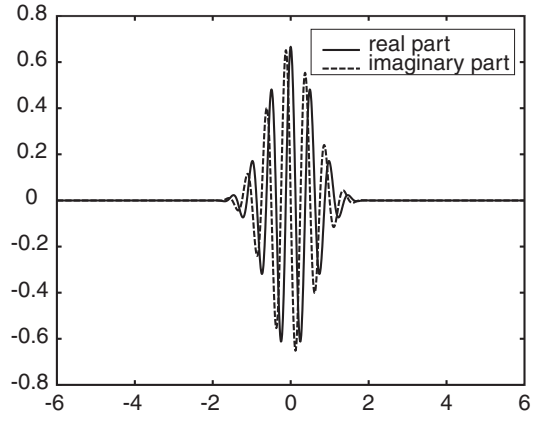


Figure A.7: Gabor-like wavelet $\beta^3(x)e^{-4\pi xj}$.

We have applied our method to the analysis of bowel movements. A magnetically active capsule was swallowed and its gastrointestinal transit was monitored. The measures consisted in its three spatial coordinates and the angles that describe its orientation [106].

Figure A.8(a) corresponds to the x-coordinate of the capsule. The sampling time was 70 ms. We have analyzed it using both the real (Figure A.8(b)) and the complex CWT (Figure A.8(c)) for cubic spline interpolation of the input signal. The y-axis corresponds to a normalized scale in seconds; it is given by $a_0 = \frac{a}{f_0}$ where f_0 is the central frequency of the wavelet.

In the real case, the wavelet was the first derivative of a quartic spline expanded by a factor of 2 as shown in Figure A.1(a). A visual inspection highlights the band of period 3 s, which corresponds to the breathing of the patient. We chose to calculate the complex CWT by an extension of the method proposed by Unser *et al*. [123] for integer scales. The analysis wavelet is the Morlet-like wavelet $\beta^3(x)e^{-4\pi xj}$ (Figure A.7). Using complex analysis (Figure A.8 (c)), we discovered three relevant frequency bands: First, the breathing with a period

close to 3 s; then, two more bands with periods around 12 s and $20 - 25$ s due to the contractions of the stomach.

A.4 Summary

We have presented a novel B-spline-based CWT algorithm that is able to compute the CWT at *any* real scale, making it possible to use arbitrary scale progressions. The computational complexity per computed coefficient is $O(1)$, as is the case with the most efficient wavelet algorithms for dyadic or integer scales. The overall operation count only depends on the wavelet shape and on the degrees of the B-spline basis on which the wavelet and the input signal are described, but is independent of the value of the scale. Thanks to the good approximation properties of B-splines, virtually any wavelet can be used (either via their B-spline interpolation or projection). Moreover, the algorithm lends itself well to a parallel implementation as it is not iterative across scales.

The price to pay for the generality of this algorithm is that the leading constant in the $O(N)$ complexity can be large (typically 56 for a cubic spline Mexican-hat wavelet and cubic interpolation of the input signal). Thus, the method really starts paying off when the size of the signal is large (say $N \geq 1000$ samples). For smaller sizes, it may be more efficient to use a simpler FFT-based method. Note, however, that the specialized version of the algorithm for integer scales beats the FFT in all cases [132].

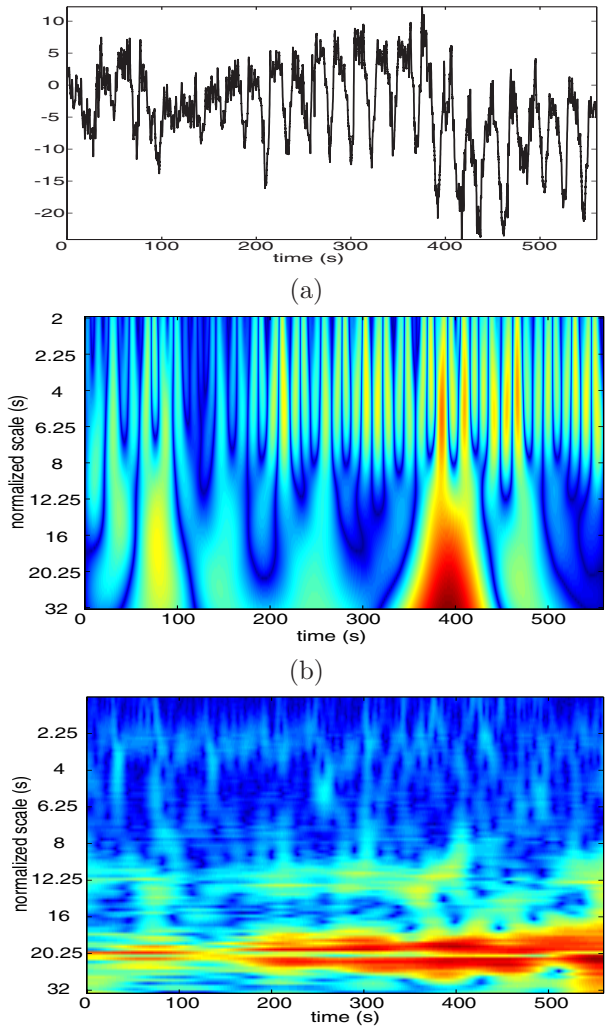


Figure A.8: (a) Trajectory of a magnet within the digestive track (x component). (b) Real CWT using the wavelet in Figure A.1(a). (c) Complex CWT using the Morlet-like wavelet shown in Figures A.7.

Bibliography

[1] E. H. Adelson, E. Simoncelli, and R. Hingorani, *Orthogonal pyramid transforms for image coding*, in Proceedings of the SPIE Conference on Visual Communication and Image Processing, 1987, pp. 50–58.

[2] M. Ahmad, E. Plotkin, M. Swamy, and J. Wang, *An iterative procedure for uniform resampling of a 2-D signal from its nonequally-spaced samples*, in Proceedings of the IEEE International Symposium on Circuits and Systems, vol. 2, 1989, pp. 1087–1090.

[3] A. Aldroubi and K. Gröchenig, *Non-uniform sampling in shift-invariant spaces*, SIAM Review, 43 (2001), pp. 585–620.

[4] A. Aldroubi and M. Unser, *Oblique projections in discrete signal subspaces of ℓ_2 and the wavelet transform*, in Proceedings of the SPIE Conference on Wavelet Applications in Signal and Image Processing II, vol. 2303, 1994, pp. 36–46.

[5] ——, *Sampling procedures in function spaces and asymptotic equivalence with Shannon's sampling theory*, Numerical Functional Analysis and Optimization, 15 (1994), pp. 1–21.

[6] ——, *Wavelets in Medicine and Biology*, New York: CRC Press, 1996.

[7] A. Aldroubi, M. Unser, and M. Eden, *Cardinal B-spline filters: Stability and convergence to the ideal lowpass filter*, Signal Processing, 28 (1992), pp. 1–12.

[8] ——, *Discrete spline filters for multiresolutions and wavelets of ℓ_2*, SIAM Journal of Mathematical Analysis, 25 (1994), pp. 1412–1432.

[9] J. P. Antoine, D. Barache, R. M. Cesar, and L. da Costa, *Multiscale shape analysis using the continuous wavelet transform*, in Proceedings of the International Conference on Image Processing, vol. 1, September 1996, pp. 291—294.

[10] G. R. Arce, Y. T. Kim, and K. E. Barner, *Order-statistic filtering and smoothing of time series: Part* 1 *and* 2., vol. 16, Elsevier Science, Amsterdam, The Netherlands, 1998.

[11] G. R. Arce and M. Tian, *Order statistic filter banks*, IEEE Transactions on Image Processing, 5 (1996), pp. 827–837.

[12] M. Asghar and E. Barner, *Nonlinear multiresolution techniques with applications to scientific visualization in a haptic environment*, IEEE Transactions on Visualization and Computer Graphics, 7 (2001), pp. 76–93.

[13] R. Bajcsy and S. Kovacic, *Multiresolution elastic matching*, Computer Graphics on Image Processing, 46 (1989), pp. 1–21.

[14] J. A. Bangham, *Properties of a series of nested median filters, namely the data sieve*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 15 (1993), pp. 31–42.

[15] J. A. Bangham, T. G. Campbell, and R. V. Aldridge, *Multiscale median and morphological filters for* 2*D pattern recognition*, Signal Processing, 38 (1994), pp. 387–415.

[16] K. Berkner and R. O. Wells, *A fast approximation to the continuous wavelet transform with applications*, in in Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers, vol. 2, 1997, pp. 135–139.

[17] T. Blu, *Iterated filter banks with rational rate change*, IEEE Transactions on Signal Processing, 41 (1993), pp. 3232–3244.

[18] T. Blu, P. Thévenaz, and M. Unser, *Minimum-support interpolators with optimum approximation properties*, in Proceedings of the IEEE International Conference on Image Processing, vol. 3, Chicago, USA, October 4–7 1998, pp. 242–245.

[19] ——, *Generalized interpolation: Higher quality at no additional cost*, in Proceedings of the IEEE International Conference on Image Processing, Kobe, Japan, October 1999, pp. 25–28.

[20] T. Blu, P. Thévenaz, and M. Unser, *MOMS: Maximal-order interpolation of minimal support*, IEEE Transactions on Image Processing, 10 (2001), pp. 1069–1080.

[21] T. Blu and M. Unser, *Approximation error for quasi-interpolators and (multi-)wavelet expansions*, Applied and Computational Harmonic Analysis, 6 (1999), pp. 219–251.

[22] T. Blu and M. Unser, *Quantitative Fourier analysis of approximation techniques: Part I—Interpolators and projectors; Part II—Wavelets*, IEEE Transactions on Signal Processing, 47 (1999), pp. 2783–2806.

[23] C. D. Boor, *On calculating with B-splines*, Journal of Approximation Theory, 6 (1972), pp. 50–62.

[24] ——, *The Spline Toolbox for Matlab*, 1997.

[25] P. Brigger, J. Hoeg, and M. Unser, *B-spline snakes: a flexible tool for parametric contour detection*, IEEE Transactions on Image Processing, 9 (2000), pp. 1484—1496.

[26] F. A. M. L. Bruekers and A. W. M. van den Enden, *New networks for perfect inversion and perfect reconstruction*, IEEE Journal on Selected Areas of Communications, 10 (1992), pp. 130—137.

[27] P. J. Burt and E. H. Adelson, *The Laplacian pyramid as a compact code*, IEEE Transactions on Communications, 15 (1986), pp. 1–21.

[28] K. Castleman, *Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1996.

[29] Y. T. Chan and K. C. Ho, *Filter design for CWT computation using the Shensa algorithm*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 3, 1999, pp. 1321–1324.

[30] G. Chang, B. Yu, and M. Vetterli, *Spatially adaptive wavelet thresholding with context modelling for image denoising*, IEEE Transactions on Image Processing, 9 (2000), pp. 1522–1531.

[31] R. Chellappa, C. L. Wilson, and S. Sirohey, *Human and machine recognition of faces: A survey*, Proceedings of the IEEE, 83 (1995), pp. 705—740.

[32] R. L. Claypoole, R. G. Baraniuk, and R. D. Nowak, *Adaptive wavelet transforms via lifting*, Department of Electrical and Computation Engineering, Rice University, Houston, Texas, Technical Report, 9304 (1999).

[33] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, *Wavelet-based statistical signal processing using hidden markov models*, IEEE Transactions on Signal Processing, 46 (1998), pp. 886–902.

[34] H. B. Curry and I. J. Schoenberg, *On pólya frequency functions IV: The fundamental spline functions and their limits.*, Journal d'Analyse Mathematique, 17 (1966).

[35] L. da F. Costa and R. M. Cesar, *Shape Analysis and Classification: Theory and Practice*, CRC Press, 2001.

[36] R. L. de Queiroz, D. A. F. Florencio, and R. W. Schafer, *Nonexpansive pyramid for image coding using a nonlinear filterbank*, IEEE Transactions on Image Processing, 7 (1998), pp. 246–252.

[37] I. Defee and Y. Neuvo, *Nonlinear filters in image pyramid generation*, in Proceedings of the IEEE International Conference on Systems Engineering, Fairborn, OH, 1991, pp. 269–272.

[38] S. Dekel and D. Leviatan, *A measure of the efficiency of convolution kernels in l_p*, Advances in Computational Mathematics, special issue on Multivariate approximation,to appear, (2002).

[39] R. A. DeVore, B. Jawerth, and B. J. Lucier, *Image compression through wavelet transform coding*, IEEE Transactions on Information Theory, 38 (1992), pp. 719–746.

[40] D. Donoho, *Denoising by shoft thresholding*, IEEE Transactions on Information Theory, 41 (1994), pp. 613–627.

[41] D. Donoho, *On minimum entropy segmentation*, in Wavelets: Theory, Algorithms, and Applications, C. K. Chui, L. Montefusco, and L. Puccio, eds., New York: Academic, 1994, pp. 233–269.

[42] D. L. Donoho and P. Y. Yu, *Nonlinear wavelet transforms based on median-interpolation*, Technical Report, (1997).

[43] O. Egger, P. Fleury, T. Ebrahimi, and M. Kunt, *High-performance compression of visual information— A tutorial review: I. Still pictures*, Proceedings of the IEEE, 87 (1999), pp. 976–1013.

[44] O. Egger and W. Li, *Very low bit rate image coding using morphological operators and adaptive decompositions*, in Proceedings of the IEEE International Conference on Image Processing, Austin, Texas, 1994, pp. 326—330.

[45] O. Egger, W. Li, and M. Kunt, *High compression image coding using an adaptive morphological subband decomposition*, Proceedings of the IEEE, 83 (1995), pp. 272–287.

[46] L. Gavit, S. Baillet, J. F. Mangin, J. Pescatore, and L. Garnero, *A multiresolution framework to MEG/EEG source imaging*, IEEE Transaction on Biomedical Engineering, 48 (2001), pp. 1080—1087.

[47] I. Gertner, *A new efficient algorithm to compute the two-dimensional discrete fourier transform*, IEEE Transactions on Acoustic, Speech and Signal Processing, 36 (1988), pp. 1036–1050.

[48] C. A. Glasbey and G. W. Horgan, *Image analysis for the biological sciences*, Wiley, West Sussex, England, 1995.

[49] H. Goely, R. D. Jones, and P. J. Bones, *Continuous wavelet transform for the detection and classification of epileptiform activity in the EEG*, in Proceedings of the First Joint BMES/EMBS Conference Serving Humanity, Advancing Technology, October 1999.

[50] I. F. Gorodnitsky and B. D. Rao, *Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm*, IEEE Transactions on Signal Processing, 45 (1997), pp. 600–616.

[51] J. Goutsias and A. M. Heijmans, *Nonlinear multiresolution signal decomposition schemes—Part I: Morphological pyramids; Part II: Morphological wavelets*, IEEE Transactions on Image Processing, 9 (2000), pp. 1862–1913.

[52] A. Gouze, M. Antonini, M. Barlaud, and B. Macq, *Optimized lifting scheme for two-dimensional quincunx sampling images*, in Proceedings of the International Conference on Image Processing, vol. 2, October 2001, pp. 253–256.

[53] F. J. Hampson and J. C. Pesquet, *M-band nonlinear subband decompositions with perfect reconstruction*, IEEE Transactions on Image Processing, 7 (1998), pp. 1547–1560.

[54] H. J. A. Heijmans, *Morphological image operators*, M.A. Academic, Boston, 1994.

[55] H. J. A. Heijmans and J. Goutsias, *Constructing morphological waveletes with the lifting scheme*, in Proceedings of the International Conference in Pattern Recognition and Image Processing, Minsk, Belarus, May 1999, pp. 65–72.

[56] N. Hekotetou and A. N. Venetsanopoulos, *Colour image interpolation for high resolution acquisition and display devices*, IEEE Transactions on Consumer Electronics, 41 (1995), pp. 1118–1126.

[57] K. C. Ho, *Fast CWT computation at integer scales by the generalized MRA structure*, IEEE Transactions on Signal Processing, 46 (1998), pp. 501–506.

[58] K. C. Ho and Y. T. Chan, *Optimum filter design for the 'à trous' algorithm*, in Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis, 1998, pp. 125–127.

[59] S. Horbelt, A. Muñoz, T. Blu, and M. Unser, *Spline kernels for continuous-space image processing*, in Proceedings of the Twenty-Fifth IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'00), vol. IV, Istanbul, Turkey, June 5-9 2000, pp. 2191–2194.

[60] H. S. Hou and H. C. Andrews, *Cubic splines for image interpolation and digital filtering*, IEEE Transactions on Acoustic, Speech and Signal Processing, 26 (1978), pp. 508–517.

[61] Z. Hu and S. T. Acton, *Morphological pyramid image registration*, in Proceedings of the 4th IEEE Southwest Symposium on Image Analysis and Interpretation, 2000, pp. 227–231.

[62] B. B. Hubbard, *The world according to wavelets*, A. K. Peters, Natick, Massachusetts, 2nd ed., 1998.

[63] D. H. Hubel and T. N. Wiesel, *Receptive fields, binocular interaction and functional architecture in the cat's visual system*, Journal of Physiology, 160 (1962), pp. 106–154.

[64] A. R. Ismail and S. S. Asfour, *Continuous wavelet transform application to EMG signals during human gait*, in Record of the Thirty-Second Asilomar Conference on Signals, Systems and Computers, vol. 1, 1998, pp. 325–329.

[65] M. Jacob, T. Blu, and M. Unser, *A unifying approach and interface for spline-based snakes*, in Proceedings of the SPIE International Symposium on Medical Imaging: Image Processing, vol. 4322, San Diego CA, USA, February 17-22 2001, pp. 340–347.

[66] B. Jeffs and M. Gunsay, *Restoration of blurred star field images by maximally sparse optimization*, IEEE Transactions on Image Processing, 2 (1993), pp. 202–211.

[67] R. Q. Jia, *Shift-invariant spaces and linear operator equations*, Israel Journal of Mathematics, 103 (1998), pp. 259–288.

[68] R. Keys, *Cubic convolution interpolation for digital image processing*, IEEE Transactions on Acoustic, Speech and Signal Processing, 29 (1981), pp. 1153–1160.

[69] J. Kovacevic and M. Vetterli, *Perfect reconstruction filterbanks with rational sampling factors*, IEEE Transactions on signal processing, 41 (1993), pp. 2047–2066.

[70] R. Kress, *Numerical Analysis*, New York: Springer, 1998.

[71] J. Kybic, P. Thévenaz, A. Nirkko, and M. Unser, *Unwarping of unidirectionally distorted EPI images*, IEEE Transactions on Medical Imaging, 19 (2000), pp. 80–93.

[72] V. D. Lecce, G. Dimauro, A. Guerriero, S. Impedoro, G. Pirlo, and A. Salzo, *Electronic document image resizing*, in Proceedings of the International Conference on Document Analysis and Recognition, Los Alamitos, CA, USA, September 20–22 1999.

[73] C. Lee, M. Eden, and M. Unser, *High-quality image resizing using oblique projection operators*, IEEE Transactions on Image Processing, 7 (1998), pp. 679–692.

[74] J. Lei, *l_p-approximation by certain projection operators*, Journal of mathematical analysis and applications, 185 (1994), pp. 1–14.

[75] J. M. Lewis and C. S. Burrus, *Approximate continuous wavelet transform with an application to noise reduction*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 3, 1998, pp. 1533–1536.

[76] S. Mallat, *Multiresolution approximation and wavelet orthogonal bases of l_2*, Transactions of the American Mathematics Society, 315 (1989), pp. 69–87.

[77] ——, *A theory of multiresolution signal decomposition: The wavelet representation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 11 (1989), pp. 674–693.

[78] S. Mallat, *Wavelets for a vision*, Proceedings of the IEEE, 4 (1996), pp. 604–614.

[79] ——, *A Wavelet Tour of Signal Processing*, Academic press, San Diego, CA, USA, 2nd ed., 1998.

[80] E. H. W. Meijering, W. J. Niessen, and M. A. Viergever, *Piecewise polynomial kernels for image interpolation: A generalization of cubic convolution*, in Proceedings of the IEEE International Conference on Image Processing, Kobe, Japan, October 25–28 1999.

[81] V. Melnik, K. Egiazarian, I. Shmulevich, and P. Kuosmanen, *A tree of median pyramidal decompositions with an application to signal denoising*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, 2000, pp. 117–120.

[82] V. Melnik, I. Shmulevich, K. Egiazarian, and J. Astola, *Iterative block-median pyramid transform-based denoising*, in Proceedings of the SPIE Conference on Wavelet Applications in Signal and Image Processing VII, Denver, CO, 1999, pp. 502–511.

[83] ——, *Block-median pyramidal transform: Analysis and denoising applications*, IEEE Transactions on Signal Processing, 49 (2001), pp. 264–372.

[84] Y. Meyer, *Advanced mathematics*, Cambridge University Press, 1992.

[85] A. Munteanu, J. Cornelis, P. D. Muynck, A. Bezerianos, and P. Cristea, *Accurate detection of coronary arteries with the continuous wavelet transform*, Computers in Cardiology, 24 (1997), pp. 601–604.

[86] C. O'Bara and S. Reisman, *Evaluation of step interpolation of systolic arterial blood pressure signals*, in Proc. IEEE 22nd. Bioengineering Conference, 1996, pp. 105–106.

[87] E. Peli, *Contrast in complex images*, Journal of the Optical Society of America, 7 (1990), pp. 2030–2040.

[88] Z. Peng, Y. F. Huang, D. J. Costello, and R. L. Stevenson, *A pyramidal image coder using generalized rank-ordered prediction filter*, IEEE Transactions on Circuits and Systems for Video Technology, 9 (1999), pp. 540–544.

[89] D. Pierre, *Optimization theory with applications*, Dover publications, New York, 1986.

[90] I. Pitas and A. N. Venetsanopolus, *Nonlinear Digital Filters: Principles and Applications*, Kuwler Academic, 1990.

[91] ——, *Order statistic in digital image processing*, Proceedings IEEE, 80 (1992), pp. 1893–1921.

[92] J. Pluim, *Mutual information based registration of medical images*, University Medical Center Utrecht, Utrecht, The Netherlands, 2000.

[93] L. Prasad, S. S. Iyengar, and S. S. Ayengar, *Wavelet analysis with applications to image processing*, CRC Press, 1997.

[94] W. K. Pratt, *Digital image processing*, Wiley, New York, 1978.

[95] C. L. R. M. Haralick, X. Zhuang and J. S. J. Lee, *The digital morphological sampling theorem*, IEEE Transactions on Acoustic, Speech and Signal Processing, 37 (1989), pp. 2067–2090.

[96] B. D. Rao and K. Kreutz-Delgado, *An affine scaling methodology for best basis selection*, IEEE Transactions on Signal Processing, 47 (1999), pp. 187–200.

[97] O. Rioul, *Fast algorithms for the continuous wavelet transform*, in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 3, 1991, pp. 2213–2216.

[98] O. Rioul, *A discrete-time multiresolution theory*, IEEE Transactions on Signal Processing, 41 (1993), pp. 2591–2606.

[99] O. Rioul and P. Duhamel, *Fast algorithm for discrete and continuous wavelet transforms*, IEEE Transactions on Information Theory, 38 (1992), pp. 569–595.

[100] O. Rioul and M. Vetterli, *Wavelets and signal processing*, IEEE Signal Processing Magazine, 8 (1991), pp. 11–38.

[101] M. Roche and S. Beusan, *A nonuniform B-spline input interpolation procedure*, Intelligent Instruments & Computers, 7 (1987), pp. 9–17.

[102] T. Rohlfing, *Efficient voxel lookup in non-uniformly spaced images using virtual uniform axes*, in Proceedings of SPIE international conference on medical imaging, vol. 4322, 2001, pp. 986–994.

[103] A. Rosenfeld, *Multiresolution image processing and analysis*, Springer, Berlin, 1984.

[104] P. Salembier and M. Kunt, *Size-sensitive multiresolution decompositions of images with rank order based filters*, Signal Processing, 27 (1992), pp. 205–241.

[105] H. Schantz, *Optical imaging and (OCR) recognition technology (recognology)*, Remittance and Document Processing Today, 14 (1992), pp. 11–15.

[106] V. Schlageter, P. Besse, R. Popovic, and P. Kucera, *Tracking system with five degrees of freedom using a 2-D arrray of Hall sensors and a permanent magnet*, Sensors and Actuators, 2951 (2001), pp. 1–6.

[107] K. Schreibman, C. Thomas, and M. Levy, *Spectral analysis of cardiac cycle length variations: Resampling overcomes effects of nonuniform spacing*, in Engineering in Medicine and Biology Society 11th. Annual International Conference, 1989.

[108] C. Segall, W. Chen, and S. Acton, *Video tracking using morphological pyramids*, Journal of Electronic Imaging, 8 (1999), pp. 176–184.

[109] J. Serra, *Image analysis and mathematical morphology*, U.K.:Academic, London, 1982.

[110] C. E. Shannon, *Communication in the presence of noise*, Proceedings of the institute of radio engineers, 37 (1949), pp. 10–21.

[111] J. L. Starck, F. Murtagh, and M. Louys, *Astronomical image compression using the pyramidal median transform*, in Astronomical Data Analysis Software and Systems IV, vol. 77, Astronomical Society of the Pacific, 1995, pp. 268–271.

[112] G. Strang, *Wavelets and dilation equations: A brief introduction*, SIAM review, 31 (1989), pp. 614–627.

[113] G. Strang and G. Fix, *A Fourier analysis of the finite-element variational method*, Constructive Aspect of Functional Analysis, (1971), pp. 796–830. Rome, Italy: Edizini Cremonese.

[114] G. Strang and T. Nguyen, *Wavelets and filter banks*, Wellesley-Cambridge, Wellesley, MY, 1996.

[115] W. Sweldens, *The lifting scheme: A new philosophy in biorthogonal wavelet constructions*, in Proceedings of SPIE Wavelet Applications Signal and Image Processing III, A. F. Laine, M. Unser, and A. Aldroubi, eds., vol. 2569, 1995, pp. 68–79.

[116] ——, *The lifting scheme: A construction of second generation wavelets*, SIAM Journal of Mathematical Analysis, 29 (1998), pp. 511–546.

[117] W. Sweldens and R. Piessens, *Asymptotic error expansions for wavelet approximations of smooth functions II*, Numerical Mathematics, 68 (1994), pp. 377–401.

[118] ——, *Quadrature formulæand aymptotic error expansions for wavelet approximations of smooth functions*, SIAM Journal of Numerical Analysis, 31 (1994), pp. 2140–2164.

[119] P. Thévenaz, T. Blu, and M. Unser, *Image Interpolation and Resampling*, Academic Press, San Diego, USA, 2000.

[120] P. Thévenaz and M. Unser, *Geometric aspects in 3D biomedical image processing*, in Bulletin ASMT/SGMT,Microvision et capteurs optiques, Journées de microtechnique, no. 22, Lausanne, Switzerland, October 1-2 1998, Association suisse de microtechnique, pp. 5–10.

[121] ——, *Optimization of mutual information for multiresolution image registration*, IEEE Transactions on Image Processing, 9 (2000), pp. 2083—2099.

[122] A. Toet, *A morphological pyramidal image decomposition*, Pattern Recognition Letters, 9 (1989), pp. 255–261.

[123] M. Unser, *Fast Gabor-like windowed Fourier and continuous wavelet transforms*, IEEE Signal Processing Letters, 1 (1994), pp. 76–79.

[124] ——, *Approximation power of biorthogonal wavelet expansions*, IEEE Transactions on Signal Processing, 44 (1996), pp. 519–527.

[125] ——, *Splines: A perfect fit for signal and image processing*, IEEE Transactions on Signal Processing, 16 (1999), pp. 22–38.

[126] M. Unser and A. Aldroubi, *A general sampling theory for nonideal acquisition devices*, IEEE Transactions on Signal Processing, 42 (1994), pp. 2915–2925.

[127] 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, 38 (1992), pp. 95–103.

[128] M. Unser, A. Aldroubi, and M. Eden, *B-spline signal processing: Part I—Theory and Part II—Efficient design and applications*, IEEE Transactions on Signal Processing, 41 (1993), pp. 834–848.

[129] ——, *A family of polynomial spline wavelet transforms*, IEEE Transactions on Signal Processing, 30 (1993), pp. 141–162.

[130] M. Unser, A. Aldroubi, and M. Eden, *The L_2 polynomial spline pyramid*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 15 (1993), pp. 364–379.

[131] M. Unser, A. Aldroubi, and M. Eden, *Enlargement or reduction of digital images with minimum loss of information*, IEEE Transactions on Image Processing, 4 (1995), pp. 247–258.

[132] M. Unser, A. Aldroubi, and S. J. Schiff, *Fast implementation of the continuous wavelet transform with integer scales*, IEEE Transactions on Signal Processing, 42 (1994), pp. 3519–3523.

[133] M. Unser and J. Zerubia, *A generalized sampling theory without band-limiting constraints*, IEEE Transactions on Circuits and Systems—II: Analog and Digital signal processing, 45 (1998), pp. 959–969.

[134] P. P. Vaidyanathan and B. Vrcelj, *Biorthogonal partners and applications*, IEEE Transactions on Signal Processing, 49 (2001), pp. 1013–1027.

[135] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*, Prentice Hall, Englewood Cliffs, NJ, 1995.

[136] M. Vrhel, C. Lee, and M. Unser, *Fractal dimension estimation using the fast continuous wavelet transform*, in Proceedings of the Wavelet Applications in Signal and Image Processing III, vol. 2569, San Diego, USA, July 12—14 1995, SPIE, pp. 478—488.

[137] M. J. Vrhel, C. L. Lee, and M. Unser, *Rapid computation of the continuous wavelet transform by oblique projections*, IEEE Transactions on Signal Processing, 45 (1997), pp. 891–900.

[138] S. A. W. Chen, *Morphological pyramids for multiscale edge detection*, in Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation, Tuscson, AZ, USA, April 1998, pp. 137–141.

[139] D. Wei and A. C. Bovik, *Sampling approximation of smooth functions via generalized coiflets*, IEEE Transactions on Signal Processing, 46 (1998), pp. 1133–1138.

[140] M. V. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software*, Wellesley, MA, 1994.

[141] H. Yoshida, B. Keserci, D. D. Casalino, O. Ozturk, A. Coskun, and A. Savranlar, *Segmentation of liver tumors in ultrasound images based on scale-space analysis of the continuous wavelet transform*, IEEE Ultrasonics Symposium, (1998), pp. 1713–1716.

Curriculum vitæ

February 5, 2002

Arrate MUÑOZ
IOA STI
Swiss Federal Institute of Technology
CH-1015 Lausanne
Switzerland
+41-(21)-693 5142 (work)
Arrate.Munoz@epfl.ch
http://bigwww.epfl.ch/munoz

Born on 16^{th} June 1973 in Pamplona, Spain. Female.

Education

Sept 1998– Present Ph.D. candidate in Biomedical Imaging Group (BIG), *Swiss Federal Institute of Technology Lausanne (EPFL)*, Switzerland. Thesis title: *"Nondyadic and Nonlinear Multiresolution Image Approximations."* Advisor: Prof. M. Unser.

Oct 97 – Jul 98 Graduate School on Communication Systems, *Swiss Federal Institute of Technology Lausanne (EPFL)*. Project:*"Scalable Resource Reservation Protocol for the Internet."* Supervisors:Dr. W. Almesberger and Prof. J.Y. Le Boudec.

Oct 96 – Jun 97 Diploma thesis, Electrical Engineering Department, *King's College London*, United Kingdom. Project: *"High Resolution*

Fibre Bragg Grating Spectrometer." Supervisor: Prof. V. Handerek.

Oct 91 – Jul 97 M.Sc. in Communication Systems Engineering, *Public University of Navarra (UPNA),* Spain.

Teaching Experience

Sept 98 – Present Teaching assistant. *Swiss Federal Institute of Technology Lausanne (EPFL).* Assistant to: *Introduction to Optics, Introduction to Image Processing and Image Processing Laboratory courses.*
Supervision of semester and graduate-school student projects.

Undergraduate Activities

Aug 97 – Sept 97 Summer research scholar. Networking group, Automation and Computation Department, *Public University of Navarra (UPNA).* Project: *"Characterization of Internet Traffic."* Supervisor: Prof. J. Aracil.

Oct 95 – Jul 96 Student research assistant with a fellowship of the Spanish Education Ministry. Spectroscopy and LASER group, Physics Department, *Public University of Navarra (UPNA).* Supervisor: Prof. J. A. Aguilera.

Graduate Activities

- Reviewer for: IEEE Transactions on Signal Processing, IEEE Transactions on Image Processing, IEEE Signal Processing Letters, Journal on Computer Vision and Image Understanding and Signal Processing Magazine.
- Co-organizer of the seminar series *"Catch a Wavelet"* on wavelet theory and applications at the *Swiss Federal Institute of Technology Lausanne (EPFL).*
- Student member of the IEEE and AEIT.

Language skills

Mother tongue Spanish, fluent English and French.

Computer skills

Languages: ANSI C, Java, Matlab, IDL, Pascal, Perl.
Software: MS Office, LaTeX, Web design and administration.
Systems: Mac OS, Linux, Win NT/98.

Other activities

I have attended lectures of **English for Management Studies** at *Westminster College*, London, United Kingdom, and **Communication as a Tool for the Engineer** at *Swiss Federal Institute of Technology (EPFL),* Switzerland.

Interested in modern art. Regular group supervision for children at summer camps.

Publications

Refereed journal papers

- A. Muñoz, T. Blu, and M. Unser, *Least-squares image resizing using finite differences*, IEEE Transactions on Image Processing, 10 (2001), pp. 1365–1378.

- A. Muñoz, T. Blu, and M. Unser, *ℓ_p-multiresolution analysis: How to reduce ringing and sparsify the error*, IEEE Transactions on Image Processing,in press.

- A. Muñoz, R. Ertle, and M. Unser, *Continuous wavelet transform with arbitrary scales and O(N) complexity*, Signal Processing Magazine, in press.

Refereed conference papers

- A. Muñoz, R. Ertlé, and M. Unser, *Fast continuous wavelet transform based on B-splines*, in Proceedings of the SPIE Conference on Mathematical Imaging: Wavelet Applications in Signal and Image Processing IX, vol. 4478, San Diego CA, USA, July 29-August 1 2001, pp. 224–229.

- A. Muñoz, T. Blu, and M. Unser, *Non-uniform to uniform grid conversion using least-squares splines*, in Proceedings of the IEEE European Conference on Signal Processing, vol. IV, Tampere, Finland, September 4–8 2000, pp. 2017–2020.

- ——, *Non-euclidean pyramids*, in Proceedings of the SPIE Conference on Mathematical Imaging: Wavelet Applications in Signal and Image Processing IX, vol. 4119, San Diego CA, USA, July 31-August 4 2000, pp. 710–720.

- S. Horbelt, A. Muñoz, T. Blu, and M. Unser, *Spline kernels for continuous-space image processing*, in Proceedings of the Twenty-Fifth IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'00), vol. IV, Istanbul, Turkey, June 5-9 2000, pp. 2191–2194.

- A. Muñoz, T. Blu, and M. Unser, *Efficient image resizing using finite differences*, in Proceedings of the 1999 IEEE International Conference on Image Processing (ICIP'99), vol. III, Kobe, Japan, October 25–28 1999, pp. 662–666.

• A. Ezbiri, A. Muñoz, S. E. Kanellopoulos, and V. A. Handerek, *High resolution fibre bragg grating sensor demodulation using a diffraction grating spectrometer and CCD detection*, in IEE Conference in Optical Techniques for Smart Structures and Structural Monitoring, February 1997, pp. 1–6.