

NONLINEAR APPROXIMATION WITH REDUNDANT MULTI-COMPONENT DICTIONARIES

THÈSE N^o 3404 (2005)

PRÉSENTÉE À LA FACULTÉ SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

Institut de traitement des signaux

SECTION DE GÉNIE ÉLECTRIQUE ET ÉLECTRONIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Lorenzo GRANAI

Laurea in Ingegneria delle Telecomunicazioni, Università degli Studi di Siena, Italie
et de nationalité italienne

acceptée sur proposition du jury:

Prof. P. Vandergheynst, directeur de thèse
Dr L. Daudet, rapporteur
Prof. M. Nielsen, rapporteur
Prof. E. Telatar, rapporteur

Lausanne, EPFL
2006

The first principle is that you must not fool yourself - and you are the easiest person to fool.

Richard P. Feynman

Acknowledgments

This dissertation has been submitted to the École Polytechnique Fédérale de Lausanne in partial fulfillment of the requirements for the degree of doctor of philosophy. I would like to express my gratitude to the members of the jury who read it carefully and contributed to its improvement with valuable comments and suggestions.

I would also like to thank to my advisor, Pierre Vandergheynst, for guiding me through the past four years. I enjoyed discussing with him and sharing ideas about research problems, but also about many others subjects. His skills and his capacity to understand people have been very precious.

Part of the work underlying this dissertation has been done in collaboration with few colleagues and students working under my supervision. Their cooperation has been important both for research purposes and for my personal development. For this reason, I would like to acknowledge the contributions of Oscar Divorra Escoda, Rosa Figueras i Ventura, Mathieu Lemay, Emilio Maggio, Javier Molinero Hernandez and Lorenzo Peotta. Moreover, I would also like to mention the helpful collaborations with Pascal Frossard, Fulvio Moschetti and Jean-Marc Vesin and the support of Alessandro Mecocci.

Thanks also to all the members of the Signal Processing Institute where I have had the opportunity to work during the last few years and in particular the director Murat Kunt.

Special thanks to all the good friends I have found in Lausanne and that make me enjoy the spare time in Switzerland! And to the ones in Italy and all over the world. Thanks to Cristina, for being close to me, helping and surprising me and for much much more!

Infine, vorrei ringraziare e dedicare questo lavoro ai miei genitori, Giuliana e Aldo. Per avermi insegnato, ascoltato e sostenuto. E sopportato!

Abstract

The problem of efficiently representing and approximating digital data is an open challenge and it is of paramount importance for many applications. This dissertation focuses on the approximation of natural signals as an organized combination of mutually connected elements, preserving and at the same time benefiting from their inherent structure. This is done by decomposing a signal onto a multi-component, redundant collection of functions (dictionary), built by the union of several subdictionaries, each of which is designed to capture a specific behavior of the signal.

In this way, instead of representing signals as a superposition of sinusoids or wavelets many alternatives are available. In addition, since dictionaries we are interested in are overcomplete, the decomposition is non-unique. This gives us the possibility of adaptation, choosing among many possible representations the one which best fits our purposes. On the other hand, it also requires more complex approximation techniques whose theoretical decomposition capacity and computational load have to be carefully studied.

In general, we aim at representing a signal with few and meaningful components. If we are able to represent a piece of information by using only few elements, it means that such elements can capture its main characteristics, allowing to compact the energy carried by a signal into the smallest number of terms.

In such a framework, this work also proposes analysis methods which deal with the goal of considering the *a priori* information available when decomposing a structured signal. Indeed, a natural signal is not only an array of numbers, but an expression of a physical event about which we usually have a deep knowledge. Therefore, we claim that it is worth exploiting its structure, since it can be advantageous not only in helping the analysis process, but also in making the representation of such information more accessible and meaningful.

The study of an adaptive image representation inspired and gave birth to this work. We often refer to images and visual information throughout the course of the dissertation. However, the proposed approximation setting extends to many different kinds of structured data and examples are given involving videos and electrocardiogram signals.

An important part of this work is constituted by practical applications: first of all we provide very interesting results for image and video compression. Then, we also face the problem of signal denoising and, finally, promising achievements in the field of source separation are presented.

Résumé

Le problème de la représentation et de l'approximation efficace de données numériques est un défi ouvert, d'une importance capitale pour de nombreuses applications. Cette thèse se concentre sur l'approximation de signaux naturels en tant que combinaison organisée d'éléments connectés, en préservant leur structure inhérente tout en en tirant parti. Ceci est accompli en décomposant un signal en une collection de fonctions (dictionnaire) redondante et possédant plusieurs composants, qui est construite par l'union de plusieurs sous-dictionnaires, chacun étant conçu pour capturer un comportement spécifique du signal.

Ainsi, plutôt que de représenter des signaux comme une superposition de sinusoïdes ou d'ondelettes, nous avons plusieurs alternatives. De plus, puisque les dictionnaires qui nous intéressent sont sur-complets, la décomposition n'est pas unique. Ceci nous offre des possibilités d'adaptation, puisque nous pouvons choisir parmi les multiples représentations possibles celle qui correspond le mieux à nos buts, mais demande aussi des techniques d'approximations plus complexes dont la capacité de décomposition théorique et la complexité doivent être étudiées avec soin.

De façon générale, nous visons à représenter un signal avec peu de composants significatifs. Si nous pouvons représenter une information en utilisant seulement quelques éléments, cela signifie que de tels éléments peuvent en capturer les caractéristiques principales, ce qui permet de comprimer l'énergie transportée par un signal en un minimum de termes.

Ce travail propose aussi des méthodes d'analyse qui permettent de considérer l'information *a priori* disponible lorsqu'un signal structuré est décomposé. En effet, un signal naturel n'est pas seulement un tableau de nombres, mais l'expression d'un événement physique dont nous avons généralement une connaissance profonde. C'est pourquoi nous maintenons qu'il est avantageux d'exploiter les relations mutuelles entre éléments constituants, car ceci peut aider non seulement le processus d'analyse mais encore à rendre la représentation d'une telle information plus accessible et significative.

L'étude d'une représentation d'image adaptative a été l'inspiration de ce travail et lui a donné naissance, c'est pourquoi nous nous référons souvent à des images et à de l'information visuelle au long de cette thèse. Cependant, ce système d'approximation s'étend à de nombreuses sortes de données structurées, et nous donnons des exemples impliquant des signaux vidéo et des électrocardiogrammes.

Une partie importante de ce travail est constituée d'applications pratiques: premièrement nous fournissons des résultats très intéressants pour la compression d'image et de vidéo. Puis, nous affrontons le problème du débruitage de signal, et finalement nous présentons des résultats prometteurs dans le domaine de la séparation de source.

Riassunto

Il problema della rappresentazione ed approssimazione efficiente dell'informazione digitale é una sfida aperta e ricopre una fondamentale importanza per molte applicazioni. Questa tesi si concentra sull'approssimazione di segnali naturali in quanto combinazione organizzata di elementi mutuamente connessi; approssimazione che preserva e allo stesso tempo beneficia della struttura inerente ai segnali. Ciò é ottenuto decomponendo un segnale su un insieme ridondante di funzioni composto da piú componenti (detto anche dizionario), creato dall'unione di diversi sotto-dizionari, ognuno dei quali é progettato per catturare uno specifico comportamento del segnale.

In questo modo, ci sono molte alternative alla rappresentazione dei segnali come superposizione di sinusoidi o wavelet. Inoltre, visto che i dizionari che ci interessano sono overcompleti la decomposizione non é unica. Questo ci fornisce la possibilitá di adattamento, scegliendo tra le molteplici possibili rappresentazioni quella che meglio si adegua ai nostri scopi. D'altra parte ciò richiede anche tecniche di approssimazione piú complesse le cui capacita' di decomposizione e il cui costo computazionale devono essere studiati attentamente.

In generale, vogliamo rappresentare un segnale con pochi e significativi componenti. Se siamo in grado di rappresentare dell'informazione usando solo pochi elementi, vuol dire che tali elementi possono catturare le sue caratteristiche principali, permettendo di compattare l'energia del segnale nel minimo numero di termini.

In questo ambito, questo lavoro propone anche metodi di analisi che mirano a considerare l'informazione *a priori* che é disponibile quando un segnale strutturato viene decomposto. Infatti, un segnale naturale non é solo una serie di numeri, bensí un'espressione di un evento fisico di cui in generale abbiamo una profonda conoscenza. Quindi sosteniamo che sfruttare la sua struttura sia vantaggioso, non solo in quanto puó aiutare il processo di decomposizione, ma anche per rendere la rappresentazione piu' accessibile e significativa.

Questo lavoro é stato ispirato dallo studio di una rappresentazione adattativa delle immagini. Nel corso di questa tesi, quindi, facciamo spesso riferimento alle immagini e all'informazione visiva. Comunque lo scenario di approssimazione qui proposto si estende a molti differenti tipi di dati strutturati e vengono forniti degli esempi che coinvolgono video e segnali cardiaci.

Una parte importante di questo lavoro é costituita da applicazioni pratiche: in primo luogo sono forniti risultati molto interessanti per quanto riguarda la compressione di immagini e video. Quindi viene anche affrontato il problema dell'eliminazione del rumore e, in fine, vengono presentati risultati promettenti nell'ambito della separazione di sorgenti.

Contents

| | |
|--|--------------|
| List of Figures | xxi |
| List of Tables | xxiii |
| 1 Introduction | 1 |
| 1.1 Roadmap | 3 |
| 2 On Approximating Images | 7 |
| 2.1 Harmonic Analysis and Approximation Theory | 7 |
| 2.2 Bidimensional Wavelets | 8 |
| 2.2.1 JPEG2000 | 9 |
| 2.3 Beyond Wavelets | 9 |
| 2.3.1 Where the *-let Family Appears and Quickly Disappears | 11 |
| 2.3.2 Other Directions | 13 |
| 2.3.3 Discussion | 14 |
| 3 An Image Model and Multi-Component Dictionaries | 15 |
| 3.1 Anisotropy and Orientation: Two Key Concepts for Edge Representation | 15 |
| 3.1.1 Anisotropic Quadtree | 16 |
| 3.1.2 Introducing Rotation | 17 |
| 3.1.3 Adding refinement | 22 |
| 3.1.4 Experimental results | 22 |
| 3.2 An Edge Oriented Redundant Dictionary | 24 |
| 3.2.1 Generating functions | 25 |
| 3.2.2 Size of the Dictionary | 27 |
| 3.3 Multi-Component Redundant Dictionaries | 28 |
| 3.3.1 A Dictionary for Textures | 31 |
| 3.3.2 MCD for Other Kinds of Structured Signals | 31 |
| 4 Function Selection Methods | 33 |
| 4.1 Sparseness | 33 |
| 4.1.1 Mathematical Setting | 34 |
| 4.1.2 Other Cost Functions | 36 |
| 4.2 Basic Original Problems | 37 |
| 4.2.1 And Their Direct Solutions | 38 |
| 4.3 Thresholding the Projections | 38 |

| | | |
|----------|--|-----------|
| 4.3.1 | Example | 39 |
| 4.4 | Greedy Algorithms | 40 |
| 4.4.1 | Matching Pursuit | 42 |
| 4.4.2 | Recovery Conditions | 43 |
| 4.4.3 | Multiple Atoms MP | 45 |
| 4.4.4 | Example | 46 |
| 4.4.5 | Use and Variations of Greedy Algorithms | 49 |
| 4.5 | Convex Relaxation Methods | 49 |
| 4.5.1 | Exact Recovery Conditions for Basis Pursuit | 49 |
| 4.5.2 | Approximation by Convex Relaxation Methods | 50 |
| 4.5.3 | A Bayesian Approach to Basis Pursuit Denoising | 50 |
| 4.5.4 | Recovery Conditions for Basis Pursuit Denoising | 51 |
| 4.5.5 | Example | 52 |
| 4.6 | Relaxed Approximation Using an L_1 Data-Fidelity Term | 54 |
| 4.6.1 | An Application to Signal Denoising | 56 |
| 4.6.2 | Special Recovery Conditions | 57 |
| 4.6.3 | Example | 58 |
| 4.7 | Other Methods | 60 |
| 4.7.1 | Method of Frames | 60 |
| 4.7.2 | Best Orthogonal Basis | 61 |
| 4.7.3 | FOCUSS | 61 |
| 5 | Image and Video Coding using Redundant Dictionaries | 63 |
| 5.1 | Still Image Compression | 64 |
| 5.1.1 | The Dictionary | 65 |
| 5.1.2 | Searching Algorithm | 65 |
| 5.1.3 | Rate-Distortion Optimization | 66 |
| 5.1.4 | Results and Comparisons | 69 |
| 5.1.5 | Reprise | 75 |
| 5.2 | Video Coding | 78 |
| 5.2.1 | Motion Estimation | 78 |
| 5.2.2 | Coding of Displaced Frame Differences | 79 |
| 5.2.3 | Dictionary Design | 80 |
| 5.2.4 | Searching Algorithm | 82 |
| 5.2.5 | Quantization and Entropy Coding | 83 |
| 5.2.6 | Rate-Distortion Optimization | 85 |
| 5.2.7 | Results and Comparisons | 87 |
| 5.3 | Discussion | 89 |
| 6 | On the use of <i>a priori</i> information for sparse signal decomposition | 91 |
| 6.1 | Mathematical Setting | 92 |
| 6.2 | Approximation by Weighted Basis Pursuit Denoising | 93 |
| 6.2.1 | Preliminary Propositions | 94 |
| 6.2.2 | Recovery Conditions of WBPDN | 95 |
| 6.2.3 | Relation with the Weighted Cumulative Coherence | 97 |
| 6.2.4 | Example: Use of Footprints and WBPDN for Sparse Approximation | 98 |
| 6.3 | Approximation by Weighted Greedy Algorithms | 100 |
| 6.3.1 | Influence of <i>a priori</i> Information on Recovery Conditions | 101 |

| | | |
|----------|--|------------|
| 6.3.2 | Rate of Convergence of Weighted-MP | 102 |
| 6.3.3 | Example: Use of Footprints and Weighted-OMP for Sparse Approximations | 104 |
| 6.4 | Natural Signal Approximation with an <i>A Priori</i> Model | 104 |
| 6.4.1 | Modeling the Relation Signal-Dictionary | 104 |
| 6.4.2 | Signal Approximation | 105 |
| 6.4.3 | Results | 106 |
| 6.5 | Including <i>a Priori</i> Information in Exact Representation Problems | 109 |
| 6.5.1 | Exact Recovery via WBP | 110 |
| 6.5.2 | Exact Recovery via Weighted-MP | 111 |
| 6.5.3 | Exact Recovery Bounds for WBP and Weighted Greedy Algorithms | 111 |
| 6.5.4 | A Toy Example for WBP in \mathbb{R}^5 | 114 |
| 6.6 | Applications | 115 |
| 6.6.1 | Images | 115 |
| 6.6.2 | Electrocardiograms | 117 |
| 6.7 | Discussion | 122 |
| 7 | Conclusion | 125 |
| 7.1 | Possible Developments | 126 |
| A | Two Proofs | 127 |
| A.1 | Proof of Lemma 6.2 | 127 |
| A.2 | Proof of Corollary 6.1 | 127 |
| | Bibliography | 129 |

List of Figures

| | | |
|------|---|----|
| 1.1 | Random collection of pixels | 1 |
| 1.2 | A natural image. From [129]. | 2 |
| 2.1 | JPEG2000 encoder building blocks. | 9 |
| 2.2 | Inadequacy of isotropic refinement for representing contours. The number of wavelets intersecting the singularity roughly doubles when the resolution increases (from left to right). In this example 6, 14 and 28 squares are used, corresponding to the 2-D wavelet coefficients. | 10 |
| 2.3 | The set of 192 basis functions obtained by the sparse coding algorithm of Olshausen and Field in [139]. | 11 |
| 2.4 | Wedgelet representation of a contour. Left: original smooth contour. Middle: wedgelet piecewise approximation. Right: wedgelet description of a leaf of the quadtree. | 12 |
| 3.1 | Example of isotropic and anisotropic quadtree decomposition. | 16 |
| 3.2 | Anisotropic and rotated quadtree of an horizon image. From top left: Original image; rotated rectangles for $j = 1, \dots, 6$; reconstruction with refinement with $M=2$ bits. | 19 |
| 3.3 | Quadtree Scheme. | 20 |
| 3.4 | Comparison among JPEG2000, isotropic quadtree with refinement and anisotropic quadtree with rotation for an image of 1024×1024 pixels. | 23 |
| 3.5 | Comparison of the RD of different curves with different Total Variation (the value of the approximation of the TV is also indicated). | 23 |
| 3.6 | Bending operation \mathcal{B}_r that arches the x_2 -axis with radius r | 25 |
| 3.7 | From left to right: a ramp and a roof edge. | 26 |
| 3.8 | Generating functions: $\phi_1(x_1, x_2)$ on the left, $\phi_2(x_1, x_2)$ on the right. | 26 |
| 3.9 | Five atoms: on the top space domain, on the bottom frequency domain represented in a logarithmic scale. The first function starting from the left is generated from ϕ_1 , the others from ϕ_2 . The effect of bending, rotating and anisotropically scaling the atom can be observed. | 27 |
| 3.10 | A natural image where we can see edges(E), textures(T) and smooth parts(S). | 28 |
| 3.11 | Laplacian Pyramid. $B(\vec{\omega})$ is the 2-D low-pass filter, $A(\vec{\omega})$ is the 2-D interpolation filter, N is the downsampling factor. | 29 |
| 3.12 | Coding scheme: W HP is the high-pass filtering using Wavelets, MP Dec and MP Rec are respectively the Matching Pursuit decomposition and reconstruction, Q represents the quantization operation, W Dec is the Wavelet decomposition, RD is the rate-distortion optimization and AC stands for Arithmetic Coding. | 30 |
| 3.13 | Examples of texture images from [12]. From the left: bark, brick wall and grass. | 31 |

| | | |
|------|---|----|
| 3.14 | Time-frequency representation of a keystroke of piano, where typical audio structures are easily visible. | 32 |
| 3.15 | On the left, an ECG signal with atrial fibrillation. It contains two physically different structures, the ventricular activity (shown in the middle), and the atrial activity (on the right). | 32 |
| 4.1 | Matrix notation: f is a column-vector of size $n \times 1$. D has size $n \times d$ and each one of its columns corresponds to an atom of the dictionary. \mathbf{c} is the coefficient column-vector of size $d \times 1$. We ask \mathbf{c} to have as much zero-elements as possible. | 35 |
| 4.2 | Capacity of ℓ_p cost functions to preserve sparseness. | 37 |
| 4.3 | Hard and soft thresholding functions. | 38 |
| 4.4 | Representation of the redundant dictionary used in this example. Each column corresponds to an atom, the first 512 columns are the CP functions, the last 128 columns are wavelet functions. | 40 |
| 4.5 | Original signal and reconstructions obtained by the thresholding method with 15 and 23 coefficients. The MSE obtained with 15 elements is 0.080, while with 23 is 0.071. | 41 |
| 4.6 | On the left: the amplitudes of the coefficients for the approximation with 15 elements. On the right: the amplitudes of the coefficients for the approximation with 23 elements. | 41 |
| 4.7 | Decomposition of a double sinusoid signal over a redundant dictionary by the Method of Frames (MOF), MP and Basis Pursuit (BP). From [33]. | 44 |
| 4.8 | Original signal and reconstructions obtained by MP with 15 and 23 coefficients. The MSE are respectively 0.016 and 0.003. | 47 |
| 4.9 | MP decomposition. On the left: the amplitudes of the coefficients for the approximation with 15 elements. On the right: the amplitudes of the coefficients for the approximation with 23 elements. | 47 |
| 4.10 | Original signal and reconstructions obtained by OMP with 15 and 23 coefficients. The MSE are respectively 0.012 and 0.003. | 48 |
| 4.11 | OMP decomposition. On the left: the amplitudes of the coefficients for the approximation with 15 elements. On the right: the amplitudes of the coefficients for the approximation with 23 elements. | 48 |
| 4.12 | Original signal and reconstructions obtained by BPDN with 15 and 23 coefficients. The MSE are respectively 0.448 and 0.107. | 53 |
| 4.13 | BPDN decomposition. On the left: the amplitudes of the coefficients for the approximation with 15 elements. On the right: the amplitudes of the coefficients for the approximation with 23 elements. | 53 |
| 4.14 | Original signal and reconstructions obtained with 15 and 23 coefficients by a BPDN decomposition followed by a projection step. The MSE are respectively 0.078 and 0.009. | 54 |
| 4.15 | BPDN decomposition followed by a projection step. On the left: the amplitudes of the coefficients for the approximation with 15 elements. On the right: the amplitudes of the coefficients for the approximation with 23 elements. | 55 |
| 4.16 | The original, noisy signal and the approximations obtained with 9 coefficients by solving (P_{1-1}) on the left and (P_{2-1}) on the right | 57 |
| 4.17 | Original signal and reconstructions obtained with 15 and 23 coefficients. Approximations are found by defining a sub-dictionary solving (P_{1-1}) and then projecting the signal onto the subspace spanned by the atoms in the sub-dictionary (see Eq. (4.45)). The MSE are respectively 0.170 and 0.077. | 59 |

| | | |
|------|--|----|
| 4.18 | Signal decomposition solving (P_{1-1}) followed by a projection step. On the left: the amplitudes of the coefficients for the approximation with 15 elements. On the right: the amplitudes of the coefficients for the approximation with 23 elements. | 59 |
| 5.1 | Encoding scheme: W HP is the high-pass filter using Wavelets, MP Dec and MP Rec are respectively the Matching Pursuit decomposition and reconstruction, Q represents the quantization operation, W Dec is the Wavelet decomposition, RD is the rate-distortion optimization and AC stands for Arithmetic Coding. | 64 |
| 5.2 | Left: Comparison between projections oriented and positions oriented coding. Right: Total number of bits per atom as a function of the number of atoms used to approximate I_{edge} . The position oriented method is used to code positions and projections. | 67 |
| 5.3 | Left: Differential entropy of the wavelet coefficients at different resolution levels. Right: bpp of our representation method as a functions of the number of atoms. The quantization step is fixed, and so the final distortion. | 68 |
| 5.4 | Comparison between JPEG2000 and MPW at about 0.05 bpp. Top: original peppers image (256×256). Bottom left: image compressed with MPW (PSNR = 23.13). Bottom right: image compressed with JPEG2000 (PSNR = 21.36). | 71 |
| 5.5 | Comparison between JPEG2000 and MPW at 0.15 (top) and 0.3 (bottom) bpp. Top left: image compressed with MPW at 0.15 bpp (PSNR = 27.03). Top right: image compressed with JPEG2000 at 0.15 bpp (PSNR = 25.89). Bottom left: image compressed with MPW at 0.3 bpp (PSNR = 29.91). Bottom right: image compressed with JPEG2000 at 0.3 bpp (PSNR = 29.79) The original image can be seen on the top of Fig. 5.4 | 72 |
| 5.6 | Comparison between JPEG2000 and MPW at 0.15 bpp. Top: original cameraman image (256×256). Bottom left: image compressed with MPW. Bottom right: image compressed with JPEG2000. | 73 |
| 5.7 | Comparison between JPEG2000 and MPW at 0.1 bpp for the image Lena of size 512×512 (on the top). Bottom left: image compressed with MPW (PSNR = 29.84). Bottom right: image compressed with JPEG2000 (PSNR = 30.00). | 74 |
| 5.8 | Laplacian Pyramid. Only the encoding part is shown. $B(\vec{\omega})$ is the 2-D low-pass filter, $A(\vec{\omega})$ is the 2-D interpolation filter, N is the downsampling factor. Q represents the quantization operation, MP the Matching Pursuit and EC stands for Entropy Coding. | 75 |
| 5.9 | Comparison between JPEG2000 and the proposed scheme based on MP at 0.079 bpp for the image Lena of size 256×256 . Left: image compressed with MP (PSNR = 25.51). Right: image compressed with JPEG2000 (PSNR = 24.32). | 76 |
| 5.10 | Comparison between JPEG2000 and the proposed scheme based on MP at 0.077 (top) and 0.147 (bottom) bpp. Top left: image compressed with MP at 0.077 bpp (PSNR = 23.92). Top right: image compressed with JPEG2000 at 0.077 bpp (PSNR = 22.62). Bottom left: image compressed with MP at 0.147 bpp (PSNR = 25.61). Bottom right: image compressed with JPEG2000 at 0.3 bpp (PSNR = 25.13). The original image can be seen on the top of Fig. 5.6 | 77 |
| 5.11 | Block diagram of a predictive video representation scheme. | 78 |
| 5.12 | A displaced frame difference output of the motion compensation of H.256/AVC. From the sequence <i>Stefan</i> in QCIF format. | 80 |
| 5.13 | Generating functions ϕ_1 and ϕ_2 | 82 |
| 5.14 | Scheme for the atom selection in the Fourier domain | 83 |

| | | |
|------|--|-----|
| 5.15 | Bytes per atom necessary to code 19 frames of <i>container</i> QCIF using different encoding styles | 84 |
| 5.16 | Example of typical bit allocations for “position” and “projection DPCM” encoding styles | 85 |
| 5.17 | Left: rate-distortion optimization: $J(m)$ for two frames of <i>Stefan</i> . Right: MSE for the first 100 frames of <i>news</i> coded with and without RD optimization | 86 |
| 5.18 | Left: MSE obtained coding the first 100 frames of <i>container</i> using MP (0.190 KBytes/frame) and the 8x8 DCT (0.194 KBytes/frame); no I-frames. Right: RD curves obtained coding the first 100 frames of <i>traffic</i> using 8x8 DCT and MP with the same motion estimation and H.264. I-frames enabled | 87 |
| 5.19 | 16th frame of the sequence <i>container</i> in QCIF format coded with MP (left) and 8x8 DCT (right) | 88 |
| 6.1 | Wavelet Footprints description scheme for a piecewise-constant signal [64]. | 98 |
| 6.2 | Dictionary formed by wavelets (left half) and its respective footprints for piecewise constant singularities (right half). | 99 |
| 6.3 | The original signal reconstructed from a 10-terms approximation computed by BPDN (left) and WBPDN (right). The comparison shows the improvement given by recomputing the projections once that the algorithm has selected a sub-dictionary. | 100 |
| 6.4 | Errors (in log scale) of the m -term approximations found by BPDN and WBPDN. On the right-hand, the approximations are computed projecting the signal onto the sub-dictionary selected by the algorithm (see Eq. (6.24)). | 100 |
| 6.5 | Comparison of OMP based approximation with 10 terms using the footprints dictionary (Fig. 6.2). Left: Original signal. Middle: “blind” OMP approximation. Right: OMP with prior knowledge of the footprints location. | 104 |
| 6.6 | Rate of convergence of the error with respect to the iteration number in the experiment of Fig. 6.5 | 105 |
| 6.7 | Experiment of approximating the 1-D signal extracted from the 140th column of <i>cameraman</i> (On the left). On the right, the rate of convergence of the residual error for OMP and Weighted-OMP. | 106 |
| 6.8 | Error (in dB) obtained by BPDN and WBPDN. Both results are obtained by using quadratic programming for selecting a dictionary subset and then recomputing the coefficients by projecting the signal onto the span of the sub-dictionary. The procedure is illustrated in Sec. 6.2.4. | 107 |
| 6.9 | Top: Approximation after 50 iterations of OMP with (right) and without (left) <i>a priori</i> information. Bottom left: Signal components captured by <i>Symmlet</i> scaling functions and Footprints using OMP. Bottom right: Signal components captured by <i>Symmlet</i> scaling functions and Footprints using Weighted-OMP. | 108 |
| 6.10 | Representation of the expectation map depending on the parameters that configure the <i>a priori</i> model in the experiment set up in Fig. 6.7. The expectation corresponds to the average energy of the residual error. | 109 |
| 6.11 | Signal $f \in \mathbb{R}^5$ to decompose over \mathcal{D} | 114 |
| 6.12 | Original image and its edginess. | 116 |
| 6.13 | Images approximated by 31 atoms selected by WBPDN (left) and BPDN (right). . . | 117 |
| 6.14 | A scheme of an ECG signal exhibiting normal sinus rhythm. | 118 |
| 6.15 | Left: QRST VA complex and its approximation using 3 atoms. Right: Effect of β on the GGF (see Eq. (6.51)). | 119 |
| 6.16 | Left: Example of a simulated AA wave during fibrillation. Right: Gabor atom. . . . | 119 |

| | | |
|------|--|-----|
| 6.17 | (a) Simulated measured 4-second ECG signal on V1. (b) Original VA on V1. (c) Estimated VA on V1 (SNR : 8.69 dB). (d) Simulated AA on V1. (e) Estimated AA on V1 (SNR : 6.81 dB) | 121 |
| 6.18 | (a) Clinical 4-second ECG signal on V2 with a dominant frequency of 1.56 Hz (see its PSD (e)). (b) Estimated VA on V2 with a dominant frequency of 1.56 Hz (see its PSD (f)). (c) Estimated AA on V2 with a dominant frequency of 7.55 Hz (see its PSD (g)). (d) Estimated AA on V2 magnified 5 times. | 122 |

List of Tables

| | | |
|-----|--|-----|
| 4.1 | MSE obtained by all the presented methods for the approximation presented in Sec. 4.3.1. | 60 |
| 5.1 | PSNR vs. bit-rate for the images <i>cameraman</i> and <i>peppers</i> : comparisons between MPW, JPEG2000 and a coding scheme based on DWT. | 70 |
| 5.2 | PSNR vs. bit-rate for the images <i>cameraman</i> and <i>Lena</i> : comparison between the proposed algorithm based on MP and JPEG2000. | 76 |
| 6.1 | Signal-to-noise ratio (dB) on lead VR, V1 and V4. The performance of our method is tested on 3 different AA amplitudes (50, 100 and 150 % of the original simulated signal). | 121 |

Introduction

A digital signal is both discrete and quantized. Therefore, a digital image is a bidimensional signal discrete in the space and in the values that it can assume. Restricting ourselves to the gray-scale case, but without losing generality, it can be seen as a set of pixels whose scalar values belong to a predefined finite range. Figure 1.1 shows a digital image that corresponds to the previous characteristics. It is a 256×256 matrix whose entries (the pixels) have an integer value which lies in $[0, 255]$. Nevertheless, Fig. 1.1 does not correspond at all to the concept we have about an image.

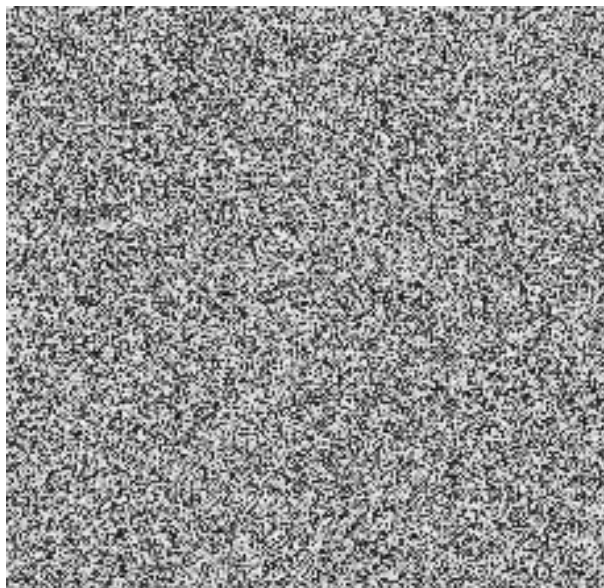


Figure 1.1: Random collection of pixels

Indeed, natural images are not random collections of pixels, as syntactically correct sentences are not random combinations of the letters of the alphabet. Borges supposes a similar situation in one of his short story [20], where he imagines a giant library which contains every single combination of the symbols of the alphabet:

...all the books, no matter how diverse they might be, are made up of the same elements: the space, the period, the comma and the twenty-two letters of the alphabet.

And again,

...the Library is total and [...] its shelves register all the possible combinations of the twenty-odd orthographical symbols (a finite number, even though extremely vast), that is all the expressible in all languages. Everything: the detailed history of the future, the archangels' autobiographies, the faithful catalogue of the Library, thousands and thousands of false catalogues...*

Similarly, the set of randomly generated matrices like the one in Fig. 1.1 can contain all the possible digital images (and therefore including the photos of my next holidays[§] - but this would drag us out of the scope of this dissertation), but has low chance to correspond to a “real world” picture.

In this work we address the problem of representing natural signals, i.e. “real world” signals, as the image in Figure 1.2. The difference between Figs. 1.1 and 1.2 is clear to everybody, notwithstanding this it is not easy to define. Interestingly, the former was obtained just by shuffling the pixels of the latter. In order to formalize such a distinction, it might be helpful to observe that an image presents peculiar elements such as edges, textures and smooth parts that are usually absent in random pixel combinations. Therefore, the problem of image representation must deal with these components or “primitives”. Note that this topic is strongly related to the characteristics of the human visual system, as it will be further discussed in the following.



Figure 1.2: A natural image. From [129].

We began this introduction speaking about images and visual information, and we will often refer to them during the rest of the dissertation. The reasons are manifold. Firstly because natural images were the starting point of the research underlying this dissertation, providing the foremost insights for a study that later has been extended to other kinds of signals. Secondly, images are used here as a paradigm of structured information, and finally they constitute one of the main fields of application for this research. In general, we are interested in catching the information contained in

*Translation from Spanish by L. Granai

§All in gray-scale and with a size of 256×256 pixels

natural signals, characterized by a structure. It is such a structure that we want to exploit in order to efficiently represent the information, and we will see how this can be done using the principle of multi-component dictionaries.

We often have the opportunity to observe how much the amount of digital information is increasing in the last years, but perhaps the most impressive phenomenon is not so much the quantity as the diversity that characterizes all this data. We have music, speech, video, text, biomedical signals like electroencephalograms and electrocardiograms, images and many many others and moreover there are collections and mixtures of them. These data types are highly structured and, at least in our mind, taking care of their own structure is a key idea for their efficient treatment and it will be the hinge of this work.

For many applications, one can be interested not much in an exact representation of a signal, as in its approximation. Moreover digital signals are usually approximation themselves, approximations of the reality. Take again the case of a numerical picture as the one in Figure 1.2: approximating it means to give an image of the image, *imago imaginis* (note that both words “image” and “representation” can be translated in Latin with the same word *imago*). In this dissertation, we will study the topic of efficient approximation, where the meaning of the word “efficient” will play a key role and it will be defined in the following. It is worth specifying that we assume to work with a digital signal, without facing the issue of how such a signal can be acquired.

In short, the problem of efficiently representing and/or approximating digital information is of key importance for many applications: starting from compression and denoising, up to image understanding (all the image processing methods that allow computers to segment images into regions, extract and classify features, recognize objects, etc.). This topic has been studied for a long time and it is significant from both theoretical and practical points of view. Furthermore, here we address the “great challenge” in image processing: finding a true and adapted way of representing images.

This dissertation lies on the edge of different scientific areas: signal processing, physics, applied mathematics, neurosciences are all involved into this work, and from all of them we try to borrow the instruments necessary for our purpose.

1.1 Roadmap

Let us now briefly present an outline of the work, along with a roadmap that illustrates our contributions. Schematically, we will face the following issues:

- Why is it interesting to use redundant approximation?
- How to use it?
- How can we better exploit signal structures?

In details, Chapter 2 presents an overview of approximation techniques, with particular emphasis on natural images. Attention is given to the recent innovations brought by computational harmonic analysis and approximation theory, whose most remarkable example is undoubtedly provided by wavelets. We then illustrate the application of wavelets to images and finally we briefly resume latest directions in image approximation, aimed at overcoming the limits of bidimensional wavelet orthogonal bases and building geometry-inspired image models.

Chapter 3 explains the importance of introducing concepts such as multiresolution, anisotropy and rotation when representing natural images. These are motivated both by our visual perception

and by mathematical analysis. Natural images, in fact, present different primitives with specific characteristics. We can easily distinguish smooth parts (low frequency components, I_{smooth}), edges (I_{edge}) and textures ($I_{texture}$). On this basis, and dealing with the previously highlighted features, we propose to model an image I as follows:

$$I \simeq I_{smooth} + I_{edge} + I_{texture}.$$

This model is therefore adopted to decompose an image by applying multiple transforms, but this implies the use of redundant collections of basis functions (also named dictionaries). In fact, such dictionaries, thanks to their flexibility, allow to better follow the plurality of behaviors shown by real world images. This leads us to introduce the concept of multi-component dictionary, i.e. an overcomplete collection of functions composed by several “sub-dictionaries”, each of them suitable for representing a specific behavior of the signal. Evidently, a multi-component dictionary issues from a multi-component data model.

We will see how this idea can be also successfully applied to any kind of structured signal. Note that the idea of multi-component dictionaries can be implemented either by using different methods in different stages of the decomposition, or by adopting some special technique to drive the analysis process. Both these strategies will be studied: an instance of the former is illustrated in Chapter 5, while the latter will be tackled in Chapter 6.

We choose to work with overcomplete dictionaries, which implies that the decomposition is non-unique. This offers us the possibility of adaptation, choosing among many possible representations the one which (most) fits our purposes. In Chapter 4 we analyze the requirements we are looking for when decomposing a signal, specifying what we mean by efficient representation or approximation, and we formally introduce the concept of sparseness. Then, we face the problem of selecting functions from a redundant dictionary, presenting some algorithms designed for this purpose. Their properties and characteristics are carefully illustrated, studying how they can be used to obtain a set of functions that provides an efficient decomposition of a given signal. This brings us to study the problem of nonlinear approximation, or better of highly nonlinear approximation, since the dictionaries we are working with are redundant.

Chapter 5 gives practical examples of approximations applied to still picture and video coding. It is shown how the image model proposed in Chapter 3, followed by a Matching Pursuit selection algorithm and an appropriate coding procedure, can give interesting and effective results in term of compression. This is especially true at low bit rates, where the algorithm is able to select the main structures of an image using very few coefficients and then compressing them efficiently. Comparisons with the standard JPEG2000 are also made.

We also illustrate here how similar decomposition techniques can be applied to video compression. Particularly, we focus on hybrid video coding schemes, representing the displaced frame difference, output of the motion compensation, by a greedy approximation over an *ad-hoc* redundant dictionary.

Chapter 6 makes a step backward. Here we develop a more general decomposition framework aimed at dealing with structured signals, and especially useful when multi-component dictionaries are adopted. We propose new algorithms that are able to take into account the *a priori* information we have about the signal we want to decompose. The properties of these algorithms are analyzed from a theoretical point of view and illustrated by examples, showing how they can lead to a better signal approximation, if the *priors* are reliable.

Finally, Chapter 7 briefly concludes the dissertation, discussing the whole work. New possible developments of this research are also presented.

On Approximating Images

This chapter illustrates the efforts spent in the last years in order to efficiently represent signals taking into account their own structure, and it pays particular attention to natural images. This is done in three steps: first, we introduce the framework of harmonic analysis, explaining the advantages of using nonlinear approximations.

Wavelets [118] are certainly the most successful technique unveiled by the recent innovation brought by harmonic analysis and approximation theory. The second part of the chapter shows how wavelets can be used in two dimensions, and how this led the design of a new standard for image compression, JPEG2000 [2, 3, 9].

Finally we show how wavelets in 2-D suffer from strong limitations and we briefly present some of the latest routes scientist are covering to approximate and compress images.

2.1 Harmonic Analysis and Approximation Theory

Harmonic analysis is the branch of mathematics which studies the representation of signals as the superposition of basic waves. These are called “harmonics”, hence the name “harmonic analysis”. In the past two centuries, it has become a vast subject with applications in areas as diverse as signal processing, quantum mechanics, and neuroscience.

Let $\{g_i\}_{i=1,2,\dots}$ be an orthonormal basis of a Hilbert space \mathcal{H} , then any $f \in \mathcal{H}$ can be written as:

$$f = \sum_{i=0}^{+\infty} \langle f, g_i \rangle \cdot g_i. \quad (2.1)$$

Instead of using all the inner products, let us consider only the first m , generating a linear approximation lying in the linear space $\mathcal{H}_m = \text{span}\{g_i : i = 1, \dots, m\}$. This subspace does not depend on f ! However, better results may be obtained by choosing the m basis functions depending on the signal: this is done by nonlinear approximation, replacing \mathcal{H}_m with a space consisting of all the elements \hat{f} in \mathcal{H} such that

$$\hat{f} = \sum_{i \in \Lambda} c_i g_i, \quad (2.2)$$

where Λ is a set of natural indexes with cardinality $|\Lambda| = m$ and the coefficients c_i are arbitrary. Here Λ does depend on f : a simple and relevant example of nonlinear approximation is given by the selection of the basis functions which have the biggest scalar products (in absolute value) with f [47]. This situation will be analyzed much more in detail in Chapter 4.

In principle, this problem corresponds to a very typical situation in science: we want to replace an object that we are not able to handle with a simpler entity (or, in this case, a linear combination of them) that is easy, or at least easier, to manipulate. Sometimes this cannot give an exact representation, or we may not need one: in this case we are looking for approximations. As just seen, nonlinear approximations give us much freedom in choosing the elements that participate to the signal expansion.

It can happen that we would like to design the set of functions g_i according to some principle: for example we may desire that they capture the structures of the signal f , or we may require them to be scalable... These requests can improve the quality of the approximation, but they can also complicate the structure of the functions g_i , preventing them to be orthogonal. So now we can ask ourselves if this is a problem. Why such set of functions should form an orthonormal basis? This last consideration opens the framework of highly nonlinear approximation, called in this way since it adds another degree of nonlinearity to the problem of function selection. The collection of functions over which we want to approximate a signal is called dictionary (\mathcal{D}). Now, supposing that $\mathcal{D} = \{g_i\}_{i \in \Omega}$ and $\Lambda \subset \Omega$, a function is decomposed as:

$$f \simeq \sum_{i \in \Lambda} c_i g_i. \quad (2.3)$$

The difference between Eqs. (2.2) and (2.3) is that in the latter Λ defines a subset of a redundant dictionary.

2.2 Bidimensional Wavelets

The recent innovations and complementary interesting points of view brought by computational harmonic analysis and approximation theory have unveiled new and powerful mathematical techniques [63]. The most successful example is undoubtedly provided by wavelets. The connection between wavelet-based coding, statistical estimation and approximation theory has shown that wavelets are optimal for estimating or compressing piecewise smooth signals with any type and number of discontinuities. This is obviously a very important class or model of signals because it describes transient behaviors, which are central to many processes. The key property of wavelets in this setting is the sparsity of the representation, i.e., the ability of wavelets to capture a very fine approximation of the signal with only few non-zero coefficients. Indeed for any such signal, the m -term nonlinear approximation error $e(m)$, that is the error measured when reconstructing the signal with the strongest m wavelet coefficients, can be shown to behave like:

$$e(m) \leq C \cdot m^{-\alpha}, \quad (2.4)$$

where α measures the smoothness of the signal [47].

Moreover a signal $f \in \mathbb{R}^n$ can be decomposed onto a wavelet basis in a very simple and fast way. This is possible since the coefficients are computed with a fast algorithm that cascades discrete convolutions of digital filters [118, 179]. Also the adoption of lifting schemes [168, 169, 179] can improve the wavelet decomposition and augment the computational efficiency. Finally the complexity results to be of the order of $O(n)$.

It was natural to use these properties in 2-D when applying wavelets to images. The extension is straightforward since any wavelet orthonormal basis of $\mathbf{L}^2(\mathbb{R})$ can be associated to a separable wavelet orthonormal basis of $\mathbf{L}^2(\mathbb{R}^2)$. 2-D decompositions can therefore be computed with a separable extension of the filter bank algorithm used in 1-D. In practice, one should simply perform a

wavelet filtering along the rows and the columns of an image or vice versa. Non-separable 2-D bases also exist, even if their use is much less frequent [107].

The applications of 2-D wavelet transform are many: in image processing it is used for contour and object detection, image retrieval, denoising, compression. In physics we can find it in astronomy and astrophysics, geophysics and fluid dynamics [16].

2.2.1 JPEG2000

The JPEG committee published its first standard for still images compression and coding in 1993*, usually known as JPEG [1, 145]. The baseline of JPEG is a Discrete Cosine Transform (DCT)-based lossy compression algorithm that uses Huffman entropy coding and operates in sequential mode. At the end of the 1990s, the JPEG committee began to investigate the possibility of creating a new still image compression standard, entirely based on the Discrete Wavelet Transform (DWT): JPEG2000 born [2, 3].

The building blocks of a typical JPEG2000 encoder are shown in Figure 2.1. In particular the compression achieved can be lossless or lossy. The DWT chosen in Part 1 of the standard is the Daubechies 9,7 for the lossy case and lifted integer-to-integer 5,3-filter bank for the lossless case. Part 2 of JPEG2000 allows for arbitrary filter specifications in the codestream.

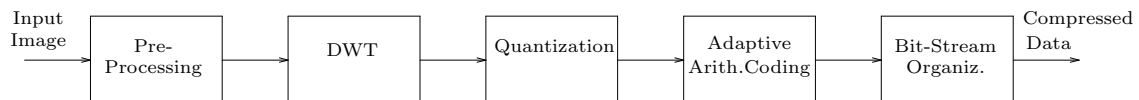


Figure 2.1: JPEG2000 encoder building blocks.

The quantization is uniform, with a deadzone that is twice the quantizer step-size (in the Part 2 of the standard the size of the deadzone may have different values for each subband). Entropy coding is performed through a context-based adaptive binary arithmetic coding. Each subband is encoded independently and partitioned into rectangular blocks. This brings many advantages, like localized random access into the image, improved functions of rotation and cropping, improved error resilience and more efficient rate control. All these benefits are obtained at expense of the exploitation of inter-subband redundancies. However, these are partly recovered by the encoding strategy.

The JPEG2000 standard exploits all the characteristics of 2-D separable wavelets, matching them with a very efficient coding strategy and a fine rate control process. The overview given here is just an hint about JPEG2000 structure and we refer a reader interested in a more accurate description to [9, 153, 170].

2.3 Beyond Wavelets

JPEG2000 is the main example of the use of wavelets for image processing. It offers many very interesting features, but it also suffers from several shortcomings [122].

We focus here on the strong limits that orthonormal wavelets show in 2-D. In fact, the efficiency of wavelets really boils down when trying to exploit the sparseness of the coefficients of natural images. Recently many researchers pointed that the wavelet transform does not give a sparse representation

*but the technical description was already frozen from 1988

of such signals, and thus might not be optimal in 2-D. Indeed, in d -dimensions the rate of nonlinear m -term approximation falls down to:

$$e(m) \leq C \cdot m^{-\frac{1}{2d-2}}. \quad (2.5)$$

This means that it does not depend on the smoothness of the signal, as it was in Equation (2.4). Moreover, this rate is only obtained for very smooth images, and there are 2-D signals where wavelets perform much worse. In other words, there is a curse of dimensions in wavelet approximation theory that really spoils the result. Figure 2.2 shows that wavelets are inefficient at representing contours because they cannot deal with their geometrical regularity. The number of coefficients needed to represent the 2-D contour increases exponentially with the resolution! One can also observe from Eq. (2.5) that in 2-D the error decays as $m^{-1/2}$, which is the same rate as Fourier expansion.

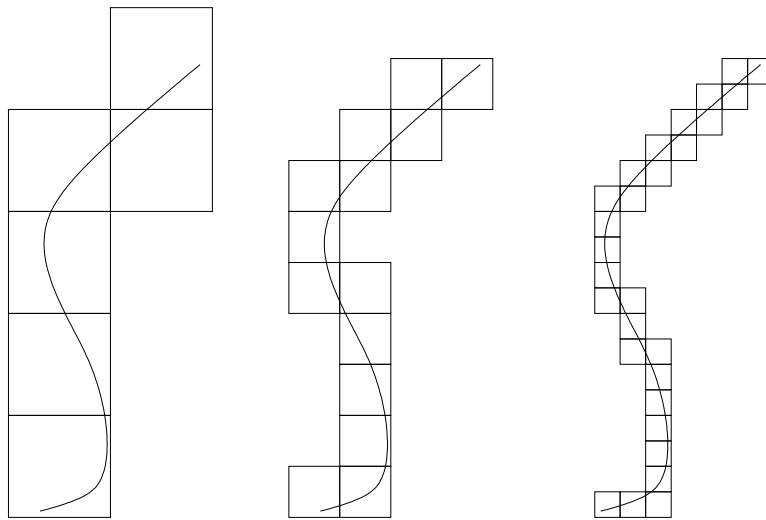


Figure 2.2: Inadequacy of isotropic refinement for representing contours. The number of wavelets intersecting the singularity roughly doubles when the resolution increases (from left to right). In this example 6, 14 and 28 squares are used, corresponding to the 2-D wavelet coefficients.

This is mainly due to the isotropic refinement: the dyadic scaling factor is applied in all directions, where clearly it should be fine along the direction of the local gradient and coarse in the orthogonal direction in order to localize the singularity in a sparse way.

In 1-D, there is only one type of singularity, point-like discontinuities in the signal or in one of its derivatives. But in higher dimensions, the geometry of possible singularities becomes extremely rich. In images, discontinuities span contours or edges. These are regular curves, whose geometric meaning is of paramount importance in describing an image. Higher dimensional wavelets completely ignore the geometry of edges and are unable to deal with curves simply because they are made from tensor products of 1-D wavelet bases and cannot cope with geometry [25]. Thus, how can we face this problem and overcome the limits of wavelets?

The issue of efficiently representing images and higher dimensional data, though of high importance, is still an open practical and theoretical challenge. An intense research activity is deployed at the border between mathematics and engineering to try to understand what the ideal representation should be. Most of the new highlights have emerged in the edge dominated problem that is the problem of representing (for data compression or statistical estimation purposes) images that are smooth away from embedded sub-manifolds (regular curves as edges).

The first answer came from a path-breaking paper of Olshausen and Field [139], where a statistical optimization experiment constructed a set of basis functions that would yield the least error when trying to approximate natural images taken from a database. The collection of these learned kernels had striking features, or more precisely a striking structure: they are organized as a multiresolution family of functions resembling Gabor atoms (see Figure 2.3). They are highly sensitive to orientation, unlike wavelets. The relevance of this result is actually twofold. First, Gabor-like basis functions seem to have been selected by natural evolution as one type of receptive fields in the human visual system [41]. It is thus very natural (and encouraging) to see them appearing in this experience. But more important even is the emergence of a structure: an orientation sensitive multiresolution representation.

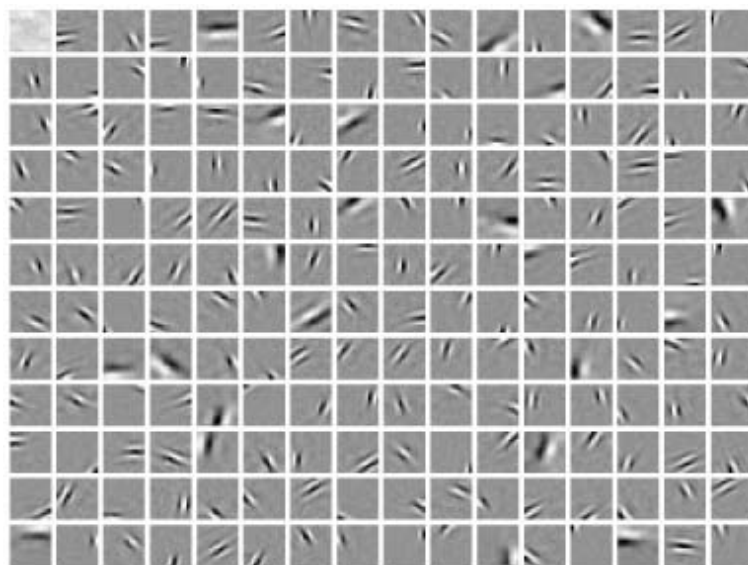


Figure 2.3: The set of 192 basis functions obtained by the sparse coding algorithm of Olshausen and Field in [139].

Later many other psycho-visual studies confirmed these striking results and pointed to the existence of a common principle involved in general sensorial information processing (e.g. see [14, 140, 141, 142, 152, 158]). Human brain represents the input sensorial information by a (relatively) small number of simultaneously active neurons. This phenomenon is commonly referred to as “sparse coding”.

Building on the preliminary results of Olshausen and Field, Donoho was one of the first to question the optimality of wavelets in higher dimensional spaces. He showed that overcomplete dictionaries of elements having particular geometric features (namely orientation sensitivity and an anisotropic scaling law) would improve on m -term nonlinear approximation [57].

2.3.1 Where the *-let Family Appears and Quickly Disappears

Many efforts were spent in the very last years in order to overcome the curse of dimensionality explained above and thus to better represent an image considering its inherent geometric structure. In this section we briefly present some of the most important and famous approaches. This list does not want to be complete, but it just aims at showing the main directions of research.

Wedgelets

Adopting the viewpoint of computational harmonic analysis, Donoho developed an overcomplete collection of functions called wedgelets [58]. They are dyadically organized with a variety of locations, scales, and orientations. Wedgelets provide nearly-optimal representations of objects in the Horizon model, as measured by minimax description length. As can be seen in Figure 2.4, wedgelets basically perform a quadtree decomposition and approximate edges by linear functions. Since they are oriented towards image coding, all the division and pruning criteria depend on a rate-distortion optimization.

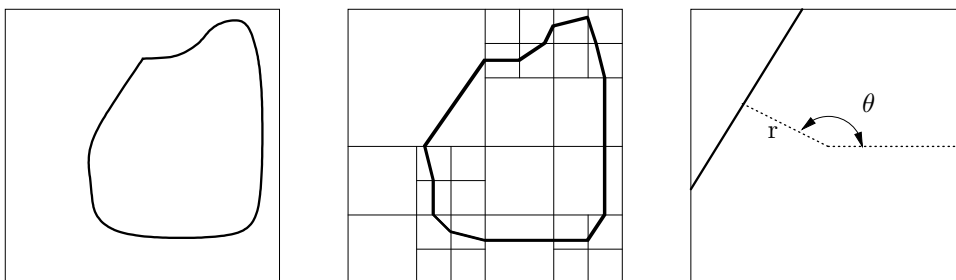


Figure 2.4: Wedgelet representation of a contour. Left: original smooth contour. Middle: wedgelet piecewise approximation. Right: wedgelet description of a leaf of the quadtree.

Curvelets

An important member of the emerging family of multiscale geometric transforms that aims at coping with image structures is the curvelet transform [25, 27]. It was developed in the last few years as an improvement of the ridgelet transform [26], in an attempt to overcome inherent limitations of traditional multiscale representations. Conceptually, the curvelet transform is a multiscale pyramid with many directions and positions at each length scale, and needle-shaped elements at fine scales. This pyramid is nonstandard, however. Indeed, curvelets have useful geometric features that set them apart from wavelets and the likes. For instance, curvelets obey a parabolic scaling relation which says that at scale 2^{-j} , each element has an envelope which is aligned along a “ridge” of length $2^{-j/2}$ and width 2^{-j} . Curvelets are interesting because they efficiently address very important problems where wavelets are far from ideal. They provide optimally sparse representations of objects which display smoothness except for discontinuity along a general curve with bounded curvature. Such representations are nearly as sparse as if the object were not singular and turn out to be far more sparse than the wavelet decomposition of the object.

This phenomenon has immediate applications in approximation theory and in statistical estimation. In approximation theory, let f_m be the m -term curvelet approximation (corresponding to the m largest coefficients in the curvelet series) to an object $f(x_1, x_2) \in \mathbf{L}^2(\mathbb{R}^2)$. Then, if f is smooth away from a singular generic smooth \mathbf{C}^2 curve, the approximation error obeys

$$\|f - f_m\|_2^2 \leq C \cdot (\log m)^3 \cdot m^{-2}.$$

Being defined in the continuous domain, the curvelet transform needs a discretization in order to be applied to digital signal. Recently a fast discrete curvelet transform has been proposed in [24]. The software package CurveLab implements the discrete curvelet transform and can be found in [7].

Curvelets can also be used as a component of a bigger dictionary. This interesting approach is followed for example in [164, 165, 166].

Contourlets

Contourlets [56] are an attractive tool for digital image processing because they are defined in the discrete domain via non-separable filter banks and consequently they are fast to compute. This construction results in a flexible multiresolution, local, and directional image expansion using contour segments. Contourlets have a tree structure and, similarly to curvelets, they impose a parabolic scale of a/a^2 . There exist several variations of the discrete contourlet transform, such as the critically sampled contourlet illustrated in [115].

Bandelets

Recently LePennec and Mallat [110] introduced a nonlinear adaptive technique for efficiently representing images by dividing the image support into regions characterized by a regular behavior. This technique achieves optimal nonlinear approximation error and, contrary to curvelets and contourlets, it is not limited to C^2 discontinuities. This approach is an attempt of exploiting the good behavior of wavelets in one dimension by a careful extension to 2-D. The basic idea of bandelets is that one can align 1-D wavelets in the direction of the edges, putting the oscillatory part of the wavelet function perpendicular to the discontinuity. Note that an analysis of the image has to be preliminary performed in order to drive the bandelet decomposition. Apparently, there exist a “second generation bandelet transform” whose preliminary description can be found in [148]. For details we refer to the web site dedicated to bandelets [6].

There are many other kinds of basis functions that belong to the *-let family: among them we can cite the beamlets [100] and the directionlets [178]. An interesting and pleasant resume of the family of *-let functions can be found in the web site in [13].

2.3.2 Other Directions

Do, Shukla et al. [55, 162] have pioneered a rate-distortion approach to the representation of edge-dominated images. Transposing the heuristic results of Donoho with coding constraints proved to outperform classical wavelet based coding on simple image models. This model, though very simple, gives a fundamental lower bound on rate-distortion analysis of edge-dominated images. This approach has led to a coding algorithm based on quadtree-structured segmentation of the image, and on a prune and merge strategy which achieves the asymptotic rate-distortion behavior predicted by the model [162]. This method can obtain very good compression results.

Another interesting approach to the problem of dealing with the representation of different features into an image is presented in [180, 181] where wavelet and wedgelet (or better *wedgeprints*, but for more details see the cited papers) basis functions are combined together in a rate-distortion compression framework.

A quite different approach has been proposed by Buccigrossi and Simoncelli in [22]. Compression is achieved by exploiting a probability model for natural images, based on empirical observation of their statistics in the wavelet transform domain.

2.3.3 Discussion

We conclude this chapter stressing again how the shortcomings shown by wavelets when applied to d -dimensional problems push many researchers to find valid alternatives for image representation. Very useful insights come from neurosciences and suggest that an efficient approximation method should take into account the geometric structures that natural images present.

We also highlight that most of the techniques previously cited renounce to the use of an orthonormal set of basis functions opening to redundant expansions. Next chapter will develop these two major points.

An Image Model and Multi-Component Dictionaries

As illustrated in Chapter 2, there is a curse of dimensions in wavelet approximation theory: higher dimensional wavelets completely ignore the geometry of images and are unable to deal with edges simply because they are made from tensor products of 1-D wavelet bases. From a rate-distortion (RD) point of view, the sub-optimality of bidimensional wavelets results in a rate-distortion decay of $O\left(R^{-\frac{1}{2}}\right)$ [118].

As a reaction to this limit of 2-D wavelets, in the last few years there has been a growing trend towards more efficient representation techniques, as explained in Chapter 2. All these attempts are inspired and led by the key concept of *geometry*.

3.1 Anisotropy and Orientation: Two Key Concepts for Edge Representation

In this section we show that a geometry-inspired representation of image contours can be efficient and helpful. Our aim is to perform a rather simple theoretical study of the asymptotic performances of anisotropic decomposition from a class of “toy” images called the “Horizon” model. This is composed of images defined on the unit square $[0, 1]^2$, such that:

$$I(x_1, x_2) = 1_{x_2 \geq y(x_1)} \quad 0 \leq x_1, x_2 \leq 1, \quad (3.1)$$

where $y(x_1) \in \mathbf{C}^p$ is p -times continuously differentiable and has finite length inside the unit square (see the image on the top left of Figure 3.2).

One way of representing this image is through a quadtree decomposition, which is in fact a toy model for wavelets. Do et al. demonstrated in [55] that the distortion D of this model decays as $D(R) \sim R^{-1}$, where the rate R is the number of bits used to code the image I , and the distortion is measured as the mean square error (MSE) between the original and the reconstructed image. The optimal quadtree is based on a dyadic division of the unit interval $[0, 1]^2$ (see Fig. 3.1(a)). At each scale, the algorithm keeps on dividing the squares containing an edge, until the maximum number of iterations J is reached, and so the maximal resolution. In [55], a refinement is performed (coding with a certain number of bits where the edge crosses the square), and then the edge is represented

by the lines that join these refinement points (so it is represented as a piecewise linear function). This approach gives a rate-distortion decay proportional to:

$$D(R) \sim \frac{\log_2 R}{R^2}. \quad (3.2)$$

This is close to the RD of wedgelets which in fact use a very similar representation scheme.

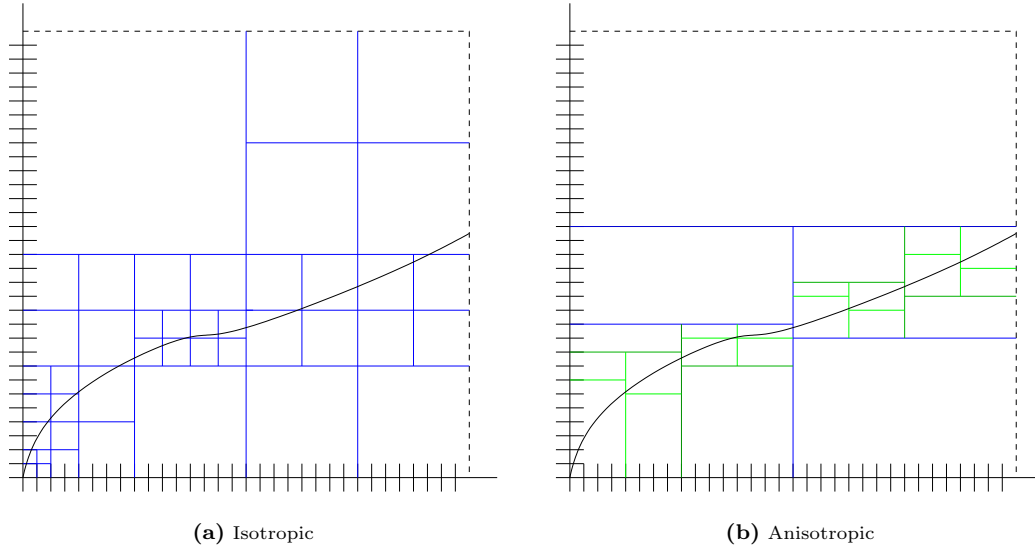


Figure 3.1: Example of isotropic and anisotropic quadtree decomposition.

Now, following [67, 68], we introduce into this quadtree scheme two key concepts: anisotropy and rotation. They will have great importance in the following of this chapter and they will result to be extremely helpful when dealing with edges. Anisotropy and rotation are used directly in the basic quadtree structure, obtaining a toy model for an adaptive nonlinear image representation tool. The fact of introducing anisotropy shows that the first derivative of the edge function appears in the rate-distortion expression. Furthermore, when rotation is also included in the scheme, the RD depends on the curvature of the edge, showing that the rate needed to represent a given contour is directly proportional to its geometrical complexity.

3.1.1 Anisotropic Quadtree

The difference between the dyadic quadtree and the anisotropic quadtree is the size of the partitions. The x_1 -axis maintains the dyadic partition, but the x_2 -axis partition never crosses the edge, as can be seen in Fig. 3.1(b). After the first partition, the rectangle containing the edge is split into two parts along the x_1 -axis. The x_2 axis is split in such a way to obtain a minimal height of the box, without crossing the edge. This process is repeated iteratively until the desired accuracy or bit-rate is reached (i.e. until the height or the width of the rectangle has the size 2^{-J}). In addition, a certain number of bits can be assigned to code the edge position in the division border, so that a straight line can approximate it. We suppose that the maximum slope of the edge inside the interval is one ($|y'(x_1)| \leq 1$), otherwise one could simply switch the axes.

Let J be the number of bits used for quantizing each of the two axes. N_{a_j} is the total number of rectangles at iteration j . There are four kind of boxes: “black”, “white”, “intermediate” and “edge”. Black and white boxes are classified to be below or above the edge, intermediate boxes contain an

edge and will be split again and edge boxes contain an edge and will not be split anymore because a sufficient precision is achieved (i.e. the edge inside the rectangle can be approximated by a straight line) or the final resolution is reached. The height of a rectangle is smaller or equal to its width times the maximum first derivative ($y'_{\max} = \max_{x_1} |y'(x_1)|$). Thus, at resolution j , the height of a box will be smaller than or equal to

$$\left\lceil \frac{2^{-j} \cdot |y'_{\max}|}{2^{-J}} \right\rceil \leq 2^{J-j} \cdot \lceil |y'_{\max}| \rceil.$$

Thus the maximum number of iterations that the quadtree will perform is:

$$j_{stop} = J + \lceil \log_2 |y'_{\max}| \rceil.$$

Notice that the case $y'_{\max} = 0$ is a special case, because it means that with only one iteration the minimum resolution has been achieved. N_e is the number of edge-rectangles and N_{a_j} is the number of all the rectangles at iteration j . Their values are:

$$N_{a_j} \sim 2^j, \quad N_e \leq 2^{J + \lceil \log_2 y'_{\max} \rceil}. \quad (3.3)$$

Since the rectangles of iteration $j + 1$ will be contained by the rectangles of the iteration j , the number of bits needed to code the size of each rectangle will decrease with j as:

$$N_{bits} \leq J - (j - 1) + \lceil \log_2 y'_{\max} \rceil. \quad (3.4)$$

Taking into account the bits needed to code whether a rectangle is black, white, edge or intermediate and the size and position, the total bit-rate needed to code the anisotropic quadtree will be given by:

$$R = (2 + N_{bits})N_a + 2M \cdot N_e, \quad (3.5)$$

where M is the number of bits used for the refinement. The first term counts the bits needed to describe the tree partition position and whether the partition is black, white, intermediate or edge, and the second term represents the bits needed to code the edge position in the finest partition. Merging (3.3) and (3.4) with (3.5), considering $M=J$ and high bit-rate and simplifying, we obtain:

$$R \sim J \cdot 2^{J + \lceil \log_2 y'_{\max} \rceil}. \quad (3.6)$$

The final distortion (at resolution 2^{-J}) is given by the sum of the distortions of all the partitions:

$$D(R) = \int_{[0,1]^2} (I - \hat{I})^2 \leq C \cdot 2^{-J-M}, \quad (3.7)$$

where \hat{I} is the reconstruction of I . When considering $M=J$ and high bit-rate, it gives $D(R) \leq C \cdot 2^{-2J}$, and from (3.7) and (3.6) we obtain:

$$D(R) \sim \frac{2^{2 \lceil \log_2 y'_{\max} \rceil} \log_2^2 R}{R^2}. \quad (3.8)$$

3.1.2 Introducing Rotation

The anisotropic quadtree shows that the edge representation can be improved in a RD sense if partitions follow the behavior of the contour. Developing this idea it is possible to use not only rectangular, but even rotated boxes. Let us take the curve in the unit interval $[0, 1]^2$ and join the

two extreme points with a line that represents its average slope. This line can be then moved up and down such that it does not cross the edge anymore. In this way we define a rectangular box that models a coarse basis function. This procedure is repeated iteratively continuing to split the x_1 axis inside the previous box in a dyadic way. Figure 3.2 shows this quadtree scheme for $j = 1, \dots, 6$. As can readily be noticed, the x_1 -partition is a fixed dyadic grid, while the x_2 -partition basically depends on the edge.

At each iteration j ($0 \leq j \leq J$) and for each box k the distortion is bounded by the area of the box that encloses the edge (see Fig. 3.3):

$$D_j^k \leq S_j^k H_j'^k = S_j^k H_j^k \cos\theta = 2^{-j} H_j^k. \quad (3.9)$$

Inside every box the edge function is approximated by its second order Taylor expansion at the central point of the partition, taking as initial partition the unit interval.

Defining x_- as the lowest point in the x_1 axis which is inside the interval to be analyzed and x_+ as the highest one, the coordinates of the two extreme points of the curve, quantized on a dyadic grid, are $(x_-, Q[y(x_-)])$ and $(x_+, Q[y(x_+)])$, where $Q[\cdot]$ stands for uniform quantization. The line joining these two points is:

$$y_{LQ}(x) = Q[y(x_-)] + \frac{Q[y(x_+)] - Q[y(x_-)]}{x_+ - x_-}(x - x_-). \quad (3.10)$$

The parallelogram cannot cross the edge, therefore its superior and inferior distances to the line are:

$$\begin{aligned} d_+ &= \max\{0, \sup(y - y_{LQ})\} \geq 0 \\ d_- &= \max\{0, \sup(y_{LQ} - y)\} \geq 0. \end{aligned} \quad (3.11)$$

Then the height of the parallelogram confining the edge is:

$$H = Q[d_+ + d_-]. \quad (3.12)$$

Three cases have to be considered: d_+ and d_- are both bigger than zero, one of them is equal to zero, and finally $d_+ = d_- = 0$. The distortion is at most the area of the parallelogram that contains the edge, as already shown in (3.9). So, when the evolution of H with the number of bits is found, the evolution of the distortion as a function of the iteration number will be known as well.

Case $d_+ > 0$ and $d_- > 0$

Let us first compute the distances d_+ and d_- in order to determine H . If x_{d_+} is the point in the x_1 axis where $y_{LQ} - y$ is maximum, we get:

$$d_+ = y(x_{d_+}) - y_{LQ}(x_{d_+}). \quad (3.13)$$

The above expression, when approximating the curve and $y(x_-)$ of Eq. (3.10) by its second order Taylor expansion at the central point of the interval being analyzed, turns to:

$$\begin{aligned} d_+ &= y\left(\frac{x_+ + x_-}{2}\right) + y'\left(\frac{x_+ + x_-}{2}\right)(x_{d_+} - \frac{x_+ + x_-}{2}) + \frac{1}{2}y''\left(\frac{x_+ + x_-}{2}\right)(x_{d_+} - \frac{x_+ + x_-}{2})^2 \\ &- y(x_-) - \frac{y(x_+) - y(x_-)}{x_+ - x_-}(x_{d_+} - x_-) + O\left(\left(x_{d_+} - \frac{x_+ + x_-}{2}\right)^3\right) \end{aligned}$$

and so:

$$\begin{aligned} d_+ &= \left[y'\left(\frac{x_+ + x_-}{2}\right) - \frac{y(x_+) - y(x_-)}{x_+ - x_-} \right] (x_{d_+} - x_-) \pm 2^{-j} \left(\frac{x_{d_+} - x_-}{x_+ - x_-} \right) \pm \frac{2^{-j}}{2} + \\ &+ \frac{1}{2}y''\left(\frac{x_+ + x_-}{2}\right) \left[\left(x_{d_+} - \frac{x_+ + x_-}{2} \right)^2 - \left(\frac{x_+ - x_-}{2} \right)^2 \right] + \\ &+ O\left(\left(x_{d_+} - \frac{x_+ + x_-}{2}\right)^3\right) + O\left(\left(\frac{x_- - x_+}{2}\right)^3\right). \end{aligned} \quad (3.14)$$

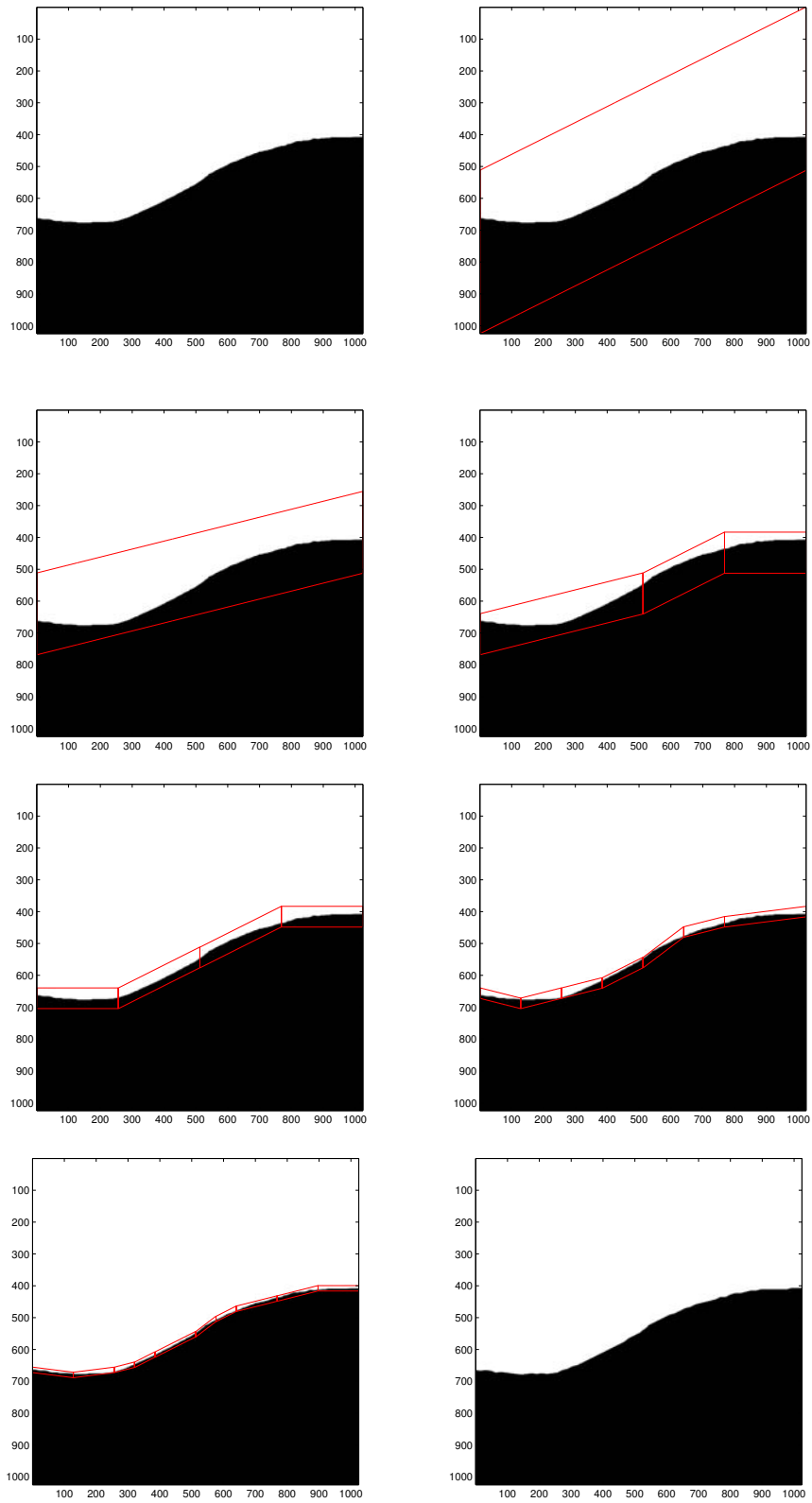


Figure 3.2: Anisotropic and rotated quadtree of an horizon image. From top left: Original image; rotated rectangles for $j = 1, \dots, 6$; reconstruction with refinement with $M=2$ bits.

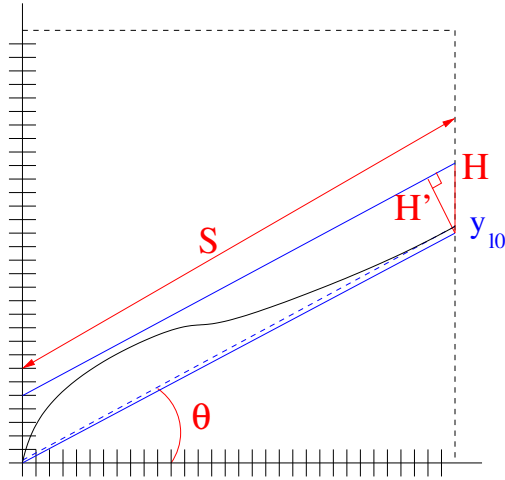


Figure 3.3: Quadtree Scheme.

It is possible to show that:

$$\left| y' \left(\frac{x_+ + x_-}{2} \right) - \frac{y(x_+) - y(x_-)}{x_+ - x_-} \right| \leq 2^{-2j} \quad (3.15)$$

and so

$$\left[y' \left(\frac{x_+ + x_-}{2} \right) - \frac{y(x_+) - y(x_-)}{x_+ - x_-} \right] (x_{d_+} - x_-) \sim O(2^{-3j}). \quad (3.16)$$

It is also easy to see that:

$$\left| \left(x_{d_+} - \frac{x_+ + x_-}{2} \right)^2 - \left(\frac{x_+ - x_-}{2} \right)^2 \right| \leq \left(\frac{2^{-j}}{2} \right)^2 = 2^{-2(j+1)}, \quad (3.17)$$

which brings the following bound for the third term in Eq. (3.14):

$$\frac{1}{2} y'' \left(\frac{x_+ + x_-}{2} \right) \left[\left(x_{d_+} - \frac{x_+ + x_-}{2} \right)^2 - \left(\frac{x_+ - x_-}{2} \right)^2 \right] \leq \frac{1}{2} \left| y'' \left(\frac{x_+ + x_-}{2} \right) \right| 2^{-2(j+1)}. \quad (3.18)$$

This expression is related to the second derivative computed in the middle point of each interval k at each iteration j . From now on, to simplify the notation, it will be referred to as K_j^k .

$$K_j^k = \left| y'' \left(\left(k + \frac{1}{2} \right) 2^{-j} \right) \right|, \quad (3.19)$$

with $0 \leq k \leq 2^j - 1$. Since the curvature of a function $y(x)$ is defined as $\frac{y''(x)}{(1+y'^2)^{\frac{3}{2}}}$, K can be considered as its approximation. As the edge is a \mathbf{C}^2 curve, the set of K_j^k is bounded:

$$\beta = \max_{0 \leq k \leq 2^j - 1} K_j^k < \infty. \quad (3.20)$$

From Eq. (3.14) it follows that the asymptotic behavior of d_+ is given by:

$$d_+ \sim \frac{1}{2} K_j^k 2^{-J - \log_2 \beta} + \frac{3}{2} \cdot 2^{-J}. \quad (3.21)$$

In fact, the other terms are $O(2^{-3j})$ (one order of magnitude smaller), so they can be neglected when computing the asymptotic behavior. Finally d_- can be found with the same computations and it shows an identical behavior.

The iterative algorithm stops when the requested resolution is reached, i.e. when $H_j = 2^{-J}$. Substituting in (3.12), the number of iterations necessary to reach this parallelogram height can be obtained as a function of the resolution and of the curvature of the edge:

$$j_{\text{stop}} = \max \left\{ 0, \frac{1}{2} (J + \log_2 \beta + 1) \right\}. \quad (3.22)$$

Now the parallelogram has width $2^{-j_{\text{stop}}}$ and height 2^{-J} , showing the a/a^2 anisotropy present in curvelets [25]:

$$\frac{\text{width}}{\text{height}} = \frac{2^{-j_{\text{stop}}}}{2^{-J}} \sim 2^{-\log_2 \beta} \frac{2^{-\frac{j}{2}}}{2^{-J}} \sim \frac{\text{width}}{\text{width}^2}, \quad (3.23)$$

The final distortion ($j = j_{\text{stop}}$) is:

$$\begin{aligned} D &= \sum_{k=0}^{2^{j_{\text{stop}}}-1} (d_+ + d_-) 2^{-j_{\text{stop}}} = \\ &= 2^{-2j_{\text{stop}}} \left(\sum_{k=0}^{2^{j_{\text{stop}}}-1} K_{j_{\text{stop}}}^k 2^{-j_{\text{stop}}} + \sum_{k=0}^{2^{j_{\text{stop}}}-1} 3 \cdot 2^{-j_{\text{stop}}} \right). \end{aligned}$$

On the right-hand side of this equation, the second sum gives a constant, while the first, when $j_{\text{stop}} \rightarrow \infty$, converges to the Riemann integral of the second derivative of the curve,

$$\lim_{j \rightarrow \infty} \sum_{k=0}^{2^{j_{\text{stop}}}-1} K_{j_{\text{stop}}}^k \cdot 2^{-j_{\text{stop}}} = \int_0^1 |y''(x)| dx, \quad (3.24)$$

which can be seen as an approximation of the Total Variation (TV) of the edge, with the only difference that we have a sum of K_j^k instead of the Riemann integral of the curvature. Naming it $\widetilde{\text{TV}}$, the final expression of the distortion turns to be:

$$D \sim (\widetilde{\text{TV}} + 3) \cdot 2^{-2j_{\text{stop}}} \sim (\widetilde{\text{TV}} + 3) \cdot 2^{-J - \log_2 \beta}. \quad (3.25)$$

Each rotated box is coded by means of H_j and a left and a right vertex. At iteration j , as at least two of the vertexes of the following parallelogram will be inside the previous one, the number of bits needed to code one vertex of the box k will evolve as:

$$N_{\text{bits}_V}^k = J - 2(j - 2) + \lceil \log_2 (K_{j-1}^k) \rceil. \quad (3.26)$$

Therefore, the total rate will be:

$$R = \sum_{j=0}^{j_{\text{stop}}} (2N_{\text{bits}_V} + N_{\text{bits}_H}) \cdot 2^j, \quad (3.27)$$

where $N_{\text{bits}_H} = N_{\text{bits}_V}$ is the number of bits needed to code the height of each box. Simplifying:

$$R \leq 3J + 2 + \sum_{j=1}^{j_{\text{stop}}} ((J - 2(j - 1) + \lceil \log_2 \beta \rceil) \cdot 3 + 2) \cdot 2^j.$$

This is an arithmetic-geometrical progression, whose sum gives (see [83]):

$$R \leq 28 \cdot 2^{\frac{1}{2}(J + \lceil \log_2 \beta \rceil)} - 3J - 26 - 6 \lceil \log_2 \beta \rceil. \quad (3.28)$$

Finally, for J big enough we can approximate the rate as:

$$R \sim 2^{\frac{1}{2}(J + \lceil \log_2 \beta \rceil)}. \quad (3.29)$$

Combining this equation with (3.25) we obtain the asymptotic RD behavior:

$$D(R) \sim (\widetilde{\text{TV}} + 3) \cdot 2^{-2 \log_2 R} \sim (\widetilde{\text{TV}} + 3) \cdot R^{-2}. \quad (3.30)$$

Case $d_+ > 0$, $d_- = 0$ or vice versa

This case turns to have the same rate-distortion behavior than the previous one, because the evolution of the rectangle height is led by the strictly positive distance.

Case $d_- = d_+ = 0$

This case is very favorable to our coding scheme, because it means that with just one iteration the minimum distortion requirement is reached. The parallelogram height will be $H = 2^{-J}$, and the rate will consist in the bits needed to code the two vertexes and the box height, $R = 3J$. This makes a RD behavior coherent with the results obtained in [55]:

$$D(R) = 2^{-\frac{R}{3}}. \quad (3.31)$$

3.1.3 Adding refinement

The anisotropic quadtree with rotation has a good RD decay, but it has the drawback that the reconstructed edge may loose its original continuity. The introduction of refinement solves this problem. This refined version of the algorithm uses for iterations from 0 to j_{stop} the same approach than in the previous case. The difference is that when the minimum resolution has been achieved, a refinement is performed inside the last resolution rectangle by splitting the x_1 axis into intervals of size 2^{-J} . The image on the bottom right of Figure 3.2 shows a reconstruction of the original horizon image performed using $J = 6$ quadtree levels and $M = 2$ bits for the refinement inside each box. The effect of adding refinement in the anisotropic quadtree with rotations does not change the slope of the RD decay, but it allows a better PSNR given a certain rate. This can be seen in Figure 3.4, where the RD line is shifted to the left by the refinement. Following the same procedure that has been previously adopted, the distortion found for the case $d_+ > 0$ and $d_- > 0$ is:

$$D \sim \widetilde{\text{TV}} \cdot 2^{-2J} + 3 \cdot 2^{-J-M}. \quad (3.32)$$

The rate now has to take into account the number of refinements performed inside each parallelogram (which is $\frac{2^{-j_{\text{stop}}}}{2^{-J}} + 1$), the number of parallelograms to refine ($2^{j_{\text{stop}}}$) and the number of bits to perform the refinement. Adding these refinement bits to (3.29) and taking $M = J$, we find:

$$R = 2^{\frac{1}{2}(J + \log_2 \beta)} + M \cdot 2^J. \quad (3.33)$$

From Equations (3.32) and (3.33), it is easy to deduce the final rate-distortion expression:

$$D(R) \sim \widetilde{\text{TV}} \cdot \frac{\log_2 R}{R^2}. \quad (3.34)$$

As in the previous section, in the case where d_+ or $d_- > 0$ nothing changes. The RD found in the case where both distances are 0 (i.e. the edge is a straight line) is very similar to the one obtained in the case without refinement.

3.1.4 Experimental results

Figure 3.4 displays a comparison among the presented methods and wavelets (JPEG2000, see Sec. 2.2.1 and [2, 8]) for a polygonal edge. These results show that the anisotropic quadtree with rotation gives better approximations than any other method. The fact that the slope for the anisotropic quadtree with or without refinement is almost the same in the graph is probably because at such low bit-rates the log factor has no influence. Even though JPEG2000 is not suited to binary images

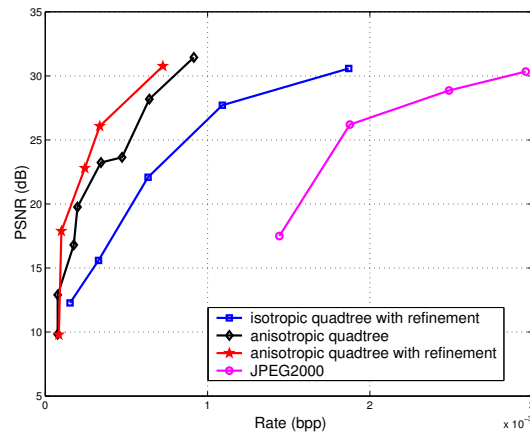


Figure 3.4: Comparison among JPEG2000, isotropic quadtree with refinement and anisotropic quadtree with rotation for an image of 1024×1024 pixels.

and so the comparison is not fair, its RD behavior is included in the graph to show that the isotropic quadtree and wavelets have really the same RD slope.

Figure 3.5 represents the rate-distortion decay of four different curves with increasing Total Variation. It shows that the practical results, obtained with the anisotropic quadtree with rotation, are coherent with the theoretical behavior found: the lower the TV, the better the RD. From left to right, the graph represents the RD of: a straight line ($TV=0$), a parabola with $TV=0.51$, a cubic curve with $TV=0.75$ and a parabola with $TV=0.89$.

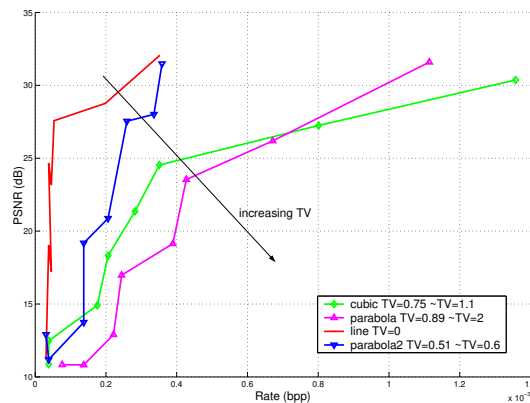


Figure 3.5: Comparison of the RD of different curves with different Total Variation (the value of the approximation of the TV is also indicated).

The fact that an approximation of the TV appears in the rate-distortion expression shows that geometrical complexity affects the capacity of compressing a given curve. As this anisotropic quadtree with rotation is a toy model for an adaptive nonlinear image representation technique, it highlights the importance of the concept of geometry. In particular, practical and theoretical results exposed in this section clearly show that the inclusion of anisotropy and rotation improves the quality of the edge decomposition (and therefore the compression).

3.2 An Edge Oriented Redundant Dictionary

In this section we design a dictionary (\mathcal{D}) aimed at representing the edges of an image. It takes into account all the previous considerations about the geometry of the contours in a natural scene. It is composed by a set of unit norm functions, also named atoms, built by translating, rotating, bending and anisotropically scaling a generating function $\phi(\vec{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}$ with $\vec{x} = (x_1, x_2)$. More precisely the four transformations are defined as follows:

a) Translation $\mathcal{T}_{\vec{b}}$, to move the atom all over the image:

$$\mathcal{T}_{\vec{b}} \phi(\vec{x}) = \phi(\vec{x} - \vec{b}). \quad (3.35)$$

b) Rotation \mathcal{R}_θ , to locally orient the atom along contours:

$$\mathcal{R}_\theta \phi(\vec{x}) = \phi(r_\theta(\vec{x})), \quad (3.36)$$

where r_θ is a rotation matrix

$$r_\theta(\vec{x}) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \quad (3.37)$$

c) Since, in general, images do not only contain straight edges, we are adapting the atoms to the shape of natural contours with a bending transformation \mathcal{B}_r . Roughly speaking, this operation arches the x_2 -axis with radius r , formally $\mathcal{B}_r \phi(\vec{x}) = \phi(\beta_r(\vec{x}))$. Figure 3.6 shows how the $\beta_r(\cdot)$ operation acts, and Figure 3.9 shows the result of bending a generating function. The transformation $\beta_r : \mathbb{R}^2 \rightarrow (-\infty, r] \times \mathbb{R}$ is not linear and it is defined as

$$\beta_r(\vec{x}) = \begin{cases} \begin{bmatrix} r - \sqrt{(x_1 - r)^2 + x_2^2} \\ r \cdot \arctan\left(\frac{x_2}{r - x_1}\right) \end{bmatrix} & \text{if } x_1 < r \\ \begin{bmatrix} r - |x_2| \\ \text{sign}(x_2) \cdot (x_1 - r + r \frac{\pi}{2}) \end{bmatrix} & \text{if } x_1 \geq r \end{cases}. \quad (3.38)$$

Applying the bending to a continuous function $\phi(\vec{x})$ we obtain $\mathcal{B}_r \phi(\vec{x})$, which is in general discontinuous on the semi-axis $[r, +\infty)$. When $\phi(\vec{x})$ is continuous and satisfies the conditions

$$\begin{aligned} \phi(r, x_2) &= \text{const} & \text{for } -r \leq x_2 \leq r & \text{ and} \\ \phi(r, x_2) &= \phi(r, -x_2) & \forall x_2 \in \mathbb{R}, \end{aligned} \quad (3.39)$$

it follows that $\mathcal{B}_r \phi(\vec{x})$ is continuous for all $\vec{x} \in \mathbb{R}^2$. The definition of the bending transformation is driven by the desire to keep the wavelet-like behavior of the generating function (see Section 3.2.1) perfectly orthogonal to the smooth direction of edges. In practice the bending transformation does not introduce discontinuities in the atoms, since the generating functions are close to zero for $x_1 = r$.

d) Anisotropic scaling \mathcal{S}_{a_1, a_2} , to adapt to contour smoothness

$$\mathcal{S}_{\vec{a}} \phi(\vec{x}) = \mathcal{S}_{a_1, a_2} \phi(x_1, x_2) = \phi\left(\frac{x_1}{a_1}, \frac{x_2}{a_2}\right). \quad (3.40)$$

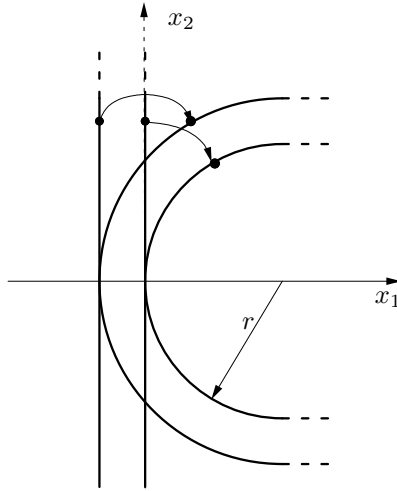


Figure 3.6: Bending operation \mathcal{B}_r that arches the x_2 -axis with radius r .

Atoms are generated varying the parameters $\vec{b}, \theta, r, \vec{a}$ of the four previous transforms in the following order:

$$g_{(\vec{b}, \theta, r, \vec{a})}(\vec{x}) = \mathcal{T}_{\vec{b}} \mathcal{R}_{\theta} \mathcal{B}_r \mathcal{S}_{\vec{a}} \phi(\vec{x}). \quad (3.41)$$

Finally the waveforms obtained are normalized:

$$g_{(\vec{b}, \theta, r, \vec{a})}^{\text{norm}}(\vec{x}) = \frac{g_{(\vec{b}, \theta, r, \vec{a})}(\vec{x})}{\|g_{(\vec{b}, \theta, r, \vec{a})}(\vec{x})\|_2}. \quad (3.42)$$

The edge-oriented dictionary can be written as in Equation (3.43), where all the parameters are discretized:

$$\mathcal{D} = \{g_{(\vec{b}, \theta, r, \vec{a})}^{\text{norm}}(\vec{x})\}_{\vec{b}, \theta, r, \vec{a}}. \quad (3.43)$$

The radius r is discretized using a dyadic grid, while for the position \vec{b} a uniform grid is kept. The two scaling factors are discretized in a uniform way. The range of the scaling factor along x_2 is bigger than the one along x_1 and it depends on the radius parameter. Moreover the scaling factor along x_2 can not exceed $\frac{\pi}{2}$ times the radius (see Fig. 3.6): the reason is that atoms that cover more than π radians are unlikely to appear in a natural image. If this is the case, one can rather use two shorter atoms, avoiding to further increase the size of the dictionary. The rotation step θ is inversely proportional to the scale a_2 . This dependency has been established because short atoms need less rotations than long ones.

3.2.1 Generating functions

The choice of the generating function $\phi(x_1, x_2)$ is driven by the idea of efficiently approximating the high frequencies of contours, like singularities in 2-D. Therefore, the atom must be a smooth low resolution function in the direction of the contour and approximate the edge transition in the orthogonal (singular) direction.

In order to be able to well represent either roof and ramp edges (see [156] and Fig. 3.7) we adopt two different generating functions, doubling in this way the size of the dictionary.

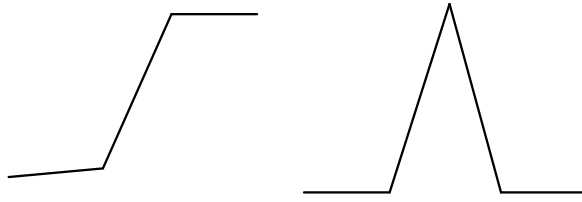


Figure 3.7: From left to right: a ramp and a roof edge.

The first function $\phi_1(\vec{x})$ is a combination of a Gaussian with its first derivative. In the x_1 -direction (which is the singular-direction) it is the first derivative of a Gaussian, while in the x_2 -direction (which is the contour-direction) it is a Gaussian, see Fig. 3.8:

$$\phi_1(x_1, x_2) = 2x_1 e^{-(x_1^2+x_2^2)}. \quad (3.44)$$

The second generating function $\phi_2(\vec{x})$, shown in Fig. 3.8, is a combination of a Gaussian and its second derivative. It was introduced in [177], motivated by the optimal joint spatial and frequency localization of the Gaussian kernel:

$$\phi_2(x_1, x_2) = (4x_1^2 - 2) \cdot e^{-(x_1^2+x_2^2)}. \quad (3.45)$$

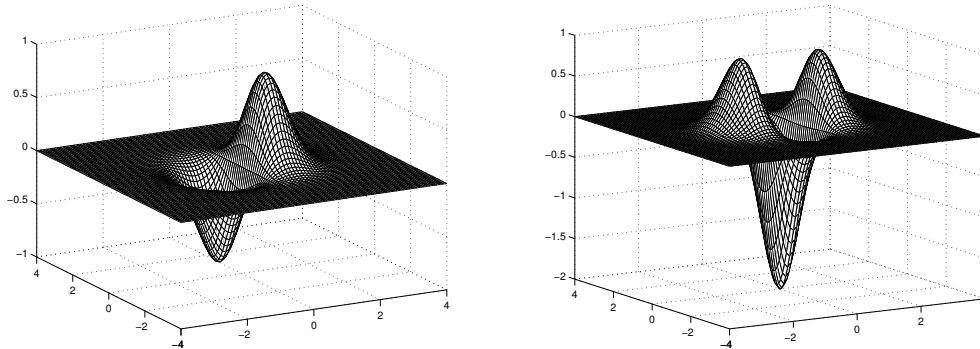


Figure 3.8: Generating functions: $\phi_1(x_1, x_2)$ on the left, $\phi_2(x_1, x_2)$ on the right.

This kind of dictionary was first introduced in [146] and then in [147]. The generating functions that appear on these papers are slightly different from the ones in Eqs. (3.44) and (3.45), In fact the Gaussian is replaced by a generalized Gaussian with shape parameter equal to four in the x_2 -direction, in order to have a faster decay in the space domain. In this way, it turns out that atoms can better approximate segments of edges and visual artifacts are reduced. The generating functions used in [147] are expressed by the following equations:

$$\begin{aligned} \phi_1(x_1, x_2) &= 2x_1 e^{-(x_1^2+x_2^4)}, \\ \phi_2(x_1, x_2) &= (4x_1^2 - 2) \cdot e^{-(x_1^2+x_2^4)}. \end{aligned} \quad (3.46)$$

Figure 3.9 shows five atoms generated using the two generating functions of (3.46) in both space and frequency domain. The effects of the transformations are also shown. It can be seen that the function ϕ_2 is more compact in the frequency domain, whereas ϕ_1 reaches lower frequencies.

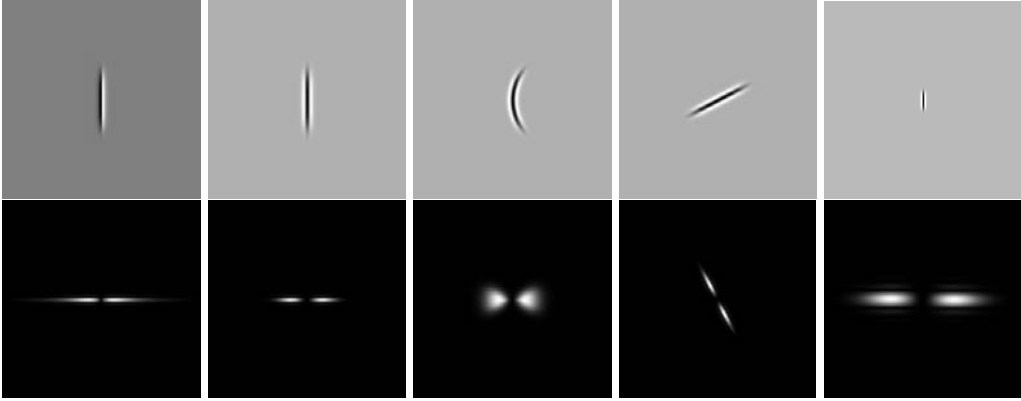


Figure 3.9: Five atoms: on the top space domain, on the bottom frequency domain represented in a logarithmic scale. The first function starting from the left is generated from ϕ_1 , the others from ϕ_2 . The effect of bending, rotating and anisotropically scaling the atom can be observed.

It is worth mentioning that these atoms present characteristics similar to the spatial receptive fields of simple cells in mammalian striate cortex. In [139] Olshausen and Field have used a learning algorithm for finding a sparse decomposition of images. They have shown how, when this algorithm is applied to a huge set of natural images (namely a high-frequency version of natural images), a set of basis functions emerges that are qualitatively similar in form to simple cell receptive fields. An example of the waveforms found using this method is shown in Figure 2.3.

These functions, as well as our atoms, are characterized by being localized in space, oriented and bandpass. An additional property of the overcomplete dictionary we propose is given by the possibility of bending the generating function. The fact that this characteristic does not appear in the waveforms obtained by Olshausen and Field can be explained by considering that the image patches they were analyzing are too small to observe such a phenomenon. However, in [105] the learning of space invariant generating functions oriented to the representation of natural images was performed using patches of 31×31 pixels, while the atoms have a support of 16×16 pixels. Such learning led to find spatially localized functions among which straight and also curved edge detectors appear.

3.2.2 Size of the Dictionary

Taking into account all the atom parameters and the two generating functions, the dictionary can be written as:

$$\mathcal{D} = \{g_{(\phi, \vec{b}, \theta, r, \vec{a})}^{\text{norm}}(\vec{x})\}_{\phi, \vec{b}, \theta, r, \vec{a}}. \quad (3.47)$$

Here $\phi \in \{\phi_1, \phi_2\}$ is the index that specifies which function has been chosen to create the atom, while the other values are the same as in Equation (3.43). Finally we obtain a highly redundant dictionary, whose size depends on the discretization of θ , r and \vec{a} . The number of rotations is chosen in proportion to the scale parameter a_2 . In general the size of the dictionary increases dramatically allowing big scaling factor along x_2 . Of course, a dictionary including elongate atoms is able to better represent long edge structures. Two instances of such a dictionary were used in [146] and [147]: they present a redundancy of about 11000 and 17000 respectively.

Even if big dictionaries can be built with a small coherence [92], our dictionary has high coherence since we adopt a geometric oriented design. The role of the coherence in redundant dictionaries will

be further analyzed in Chapter 4.

3.3 Multi-Component Redundant Dictionaries

Edges are certainly the most visually relevant elements of images, and this justifies the efforts that many scientists spent in order to find an effective way to represent them. However, natural images are not only composed by edges, but they present different primitives with peculiar characteristics. As shown by Figure 3.10, we can easily distinguish smooth parts (low frequency components, I_{smooth}), edges (I_{edge}) and textures ($I_{texture}$). Formally, we propose here the following model for an image I :

$$I \simeq I_{smooth} + I_{edge} + I_{texture}. \quad (3.48)$$

The sign “+” simply states that I_{smooth} and I_{edge} and $I_{texture}$ are present in a natural image. Nevertheless, making the hypothesis that the three components do not overlap (which is quite reasonable) also allows to give to the “+” in Eq. (3.48) a mathematical meaning, i.e. the sum of the pixel values.



Figure 3.10: A natural image where we can see edges(E), textures(T) and smooth parts(S).

In the previous section we have proposed a dictionary designed for I_{edge} . What one needs now are functions that can well represent the textures and the smooth parts, and a model for the decomposition.

At first, we focus our attention on the smooth part. In [146] we proposed a decomposition scheme aimed at compressing an image at a very low bit-rate. In these conditions the most visually relevant components are included in I_{smooth} and I_{edge} , allowing us to use the representation method illustrated in Figure 3.11, where an image is decomposed into low frequencies and high frequencies using the laplacian pyramid scheme of Burt and Adelson [23]. From an original image, the laplacian pyramid scheme derives a coarse approximation by low-pass filtering and downsampling. Based on this coarse version, it predicts the original by upsampling and filtering and calculates the difference as the high-pass version or detail version. The detail version is then represented using the atoms from the edge-oriented dictionary.

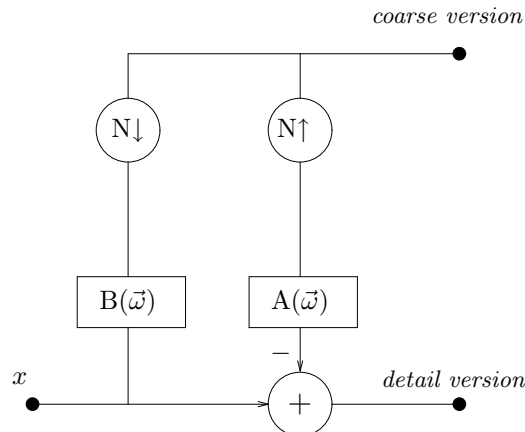


Figure 3.11: Laplacian Pyramid. $B(\vec{\omega})$ is the 2-D low-pass filter, $A(\vec{\omega})$ is the 2-D interpolation filter, N is the downsampling factor.

The filters $B(\vec{\omega})$ and $A(\vec{\omega})$ are set to be the same for sake of simplicity, even though better coding results may be obtained by choosing the two filters independently. The filtering process is performed applying three times a low-pass filter followed by downsampling. The filter used is a 11×11 taps, symmetric low-pass FIR filter, designed using the window method. The window used is Gaussian with variance $\sigma^2 = 2$ pixels, the normalized cut-off frequency is 0.45, while the downsampling factor is two.

With such a simple method we first separate the two image components and then we code them with an appropriate technique. In [146], using such a scheme for compression, the coarse version of the signal was uniformly quantized and entropy coded, while the detail version was decomposed over the edge-oriented dictionary using the Matching Pursuit algorithm. Details on this algorithm will be given in Chapter 5.

Such a model can be modified and upgraded in order to take also textures into account. This is done, for example, by the image representation scheme presented in [147] and illustrated in Fig. 3.12. Also this scheme is designed for image compression and it will be used by the coding algorithm presented in Chapter 5. At the present moment we just focus our attention on the decomposition that it adopts. The dictionary, described in Section 3.2, has been designed to match the object contours, whose energy is mostly localized at high frequencies. Therefore, before coding the edges, the image is decomposed with wavelets and reconstructed keeping all the subbands but the low-pass. This step is equivalent to a high-pass filtering and it is labeled as “W HP” in Fig 3.12. The high frequency content of the signal is thus decomposed over the edge-oriented dictionary using again the Matching Pursuit algorithm for selecting the functions. After that the coefficients have been quantized, a residual image is computed by subtracting the quantized reconstruction from the original input image. This residual contains the low frequencies of the signal, the textures and the artifacts introduced by Matching Pursuit (the latter also include quantization errors). As can be seen in Fig. 3.12 the residual is decomposed with wavelets. The wavelet functions, used for both decomposing the residual and computing the high-pass input for Matching Pursuit, are the Cohen-Daubechies-Feauveau 9,7 [35]. At this point we have low-pass wavelet coefficients (projection on the scaling function) representing I_{smooth} , atoms from the edge-oriented dictionary representing I_{edge} and the high frequency wavelet coefficients representing $I_{texture}$ and correcting the Matching Pursuit artifacts, if any.

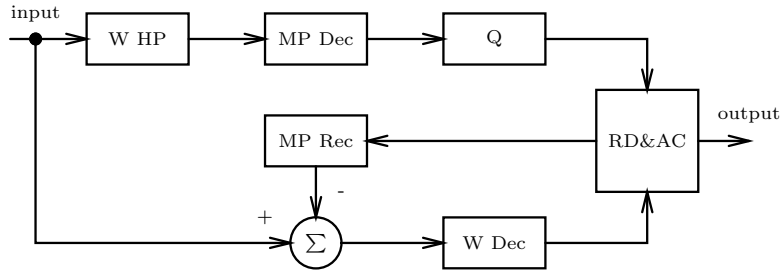


Figure 3.12: Coding scheme: W HP is the high-pass filtering using Wavelets, MP Dec and MP Rec are respectively the Matching Pursuit decomposition and reconstruction, Q represents the quantization operation, W Dec is the Wavelet decomposition, RD is the rate-distortion optimization and AC stands for Arithmetic Coding.

More informations about the two previous compression schemes, as well as practical examples and results, will be copiously given in Chapter 5.

The reader can observe how the image model of Equation (3.48) is reflected by the decomposition method of Figure 3.12. Of course, this is just an example and many other possibilities are available. Far from making an exhaustive list of all the works fitting this framework, we can however observe that many researches investigate on this direction. In particular here we want to cite the pioneering study of Yves Meyer [123] describing the so-called $u + v$ models, where the u component is aimed at modeling the objects or important features and the v component represents textures and noise. On these line see also [131] and [157]. Wakin et al. in [180, 181] propose another technique that aims at compressing still images using a similar model. They make use of wedgeprints (anisotropic atoms that are adapted to edge singularities, derived from Donoho's wedgelets [58]) for coding the edges and wavelets for the other image components. Also bandelets [110] can be interpreted in such framework, since they are adapted to the image geometry. Other works in this direction can be found in [164, 165] and in the very recent work on morphological component analysis [166].

In general, given the image model of Eq. (3.48), one needs a decomposition scheme as the two ones illustrated in Figures 3.11 and 3.12. Such schemes imply the use of a Multi-Component Dictionary (MCD), that we can define as a large, redundant collection of functions built by the union of $q \geq 2$ sub-dictionaries \mathcal{D}_j , each of which is particularly appropriate for describing a given class of features:

$$\mathcal{D} = \bigcup_j \mathcal{D}_j, \text{ with } 1 \leq j \leq q. \quad (3.49)$$

Let us call g_i the unit norm atoms that compose $\mathcal{D} = \{g_i\}_{i \in \Omega}$. Therefore, the expansion of a signal f over a MCD looks like (see also [85]):

$$f = \sum_{j=1}^q \sum_{i \in \Lambda_j} c_i \cdot g_i, \quad (3.50)$$

where $\Lambda_j \subset \Omega$ are subsets that index the basis functions of the sub-dictionary \mathcal{D}_j and such that $\bigcup_{j=1}^q \Lambda_j = \Omega$. Of course, c_i are the coefficients of the functions g_i .

In the case of the image model we propose in (3.48), one should need $q = 3$ sub-dictionaries, one for I_{edge} , one for I_{smooth} , and one for $I_{texture}$. The first issue has already been widely faced in Section 3.2, ending up with an edge-oriented collection of functions.

Concerning the case of I_{smooth} , we gave two examples in this Section, using either low-pass functions as in Fig. 3.11, either wavelets scaling functions. Another possibility used in [69] and [85]

is to use a sub-dictionary composed by translated and isotropically scaled versions of a Gaussian:

$$g(x_1, x_2) = \frac{1}{\pi} e^{-(x_1^2 + x_2^2)}. \quad (3.51)$$

The transformations, are very similar to the ones defined in Section 3.2, Eqs. (3.35), and (3.40), except that the scaling factor is the same for both axes, and of course rotation and bending are missing.

3.3.1 A Dictionary for Textures

Roughly speaking, texture images are specially homogeneous and consist of repeated elements, often subject to some randomization in their location, size, color, orientation, etc. Figure 3.13 shows some examples of gray-scale textures.

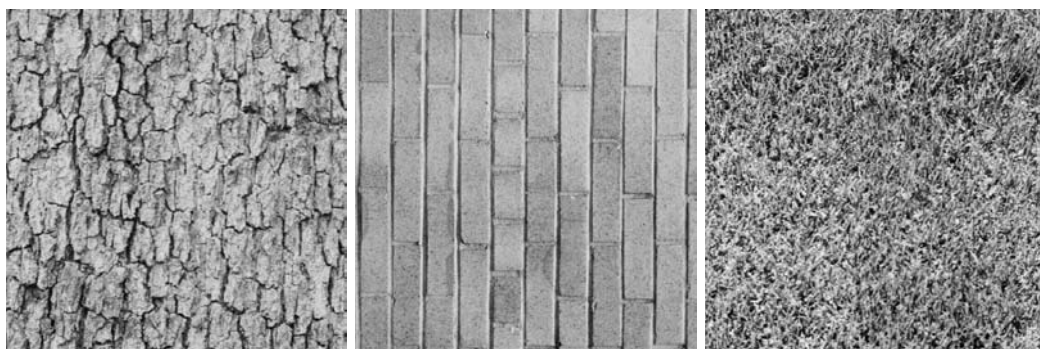


Figure 3.13: Examples of texture images from [12]. From the left: bark, brick wall and grass.

Julesz pioneered the statistical characterization of textures hypothesizing that the n th-order joint empirical densities of image pixels (for some unspecified n), could be used to divide textures into classes, indistinguishable to a human observer [106]. Since then, many different texture models were proposed, but is still extremely difficult to design a dictionary that can well represent these kind of primitives. Many interesting researches aim at building statistical texture models, mainly using the theory of Markov random fields and/or oriented linear kernels at multiple spatial scales. Within this category we can cite the excellent approach of Portilla and Simoncelli for parametric texture modeling (and also for synthesis) using wavelets coefficients [150, 151].

The method we presented in [147] makes use of wavelet functions to code textures (see the scheme in Fig. 3.12), obtaining good results. Another simple and quite efficient basis can be given by the classical Discrete Cosine Transform (DCT), applied to the whole image [165], or better spatially localized (local DCT).

3.3.2 MCD for Other Kinds of Structured Signals

To conclude this chapter, we would like to stress how the principle of multi-component dictionaries can be extended to many signals other than natural images. Just suppose we have a structured signal, for which we can individuate some class of important features. These can come from the information we have about the physical background of the signal, rather than from empirical observation. Suppose also to have a signal model, based on these components. If one is able to design a dictionary, namely a sub-dictionary, that well catches the main characteristics of each class of features, we can easily use MCD for a “structure-inspired” decomposition.

There exist plenty of signal families that present the required characteristics. One example is given by audio, and more specifically speech and music [19, 40, 87, 90, 126]. Following for instance [127] or [88], one can individuate in an audio signal transient and tonal parts that can play the role of edges and smooth parts in natural images (but with the important difference that they may overlap). Figure 3.14 illustrates a time-frequency representation of a keystroke of piano obtained through a Matching Pursuit decomposition with 5000 Gabor atoms [88, 89]. One can easily distinguish the string (long horizontal structures) from the hammer (vertical structure, the transient).

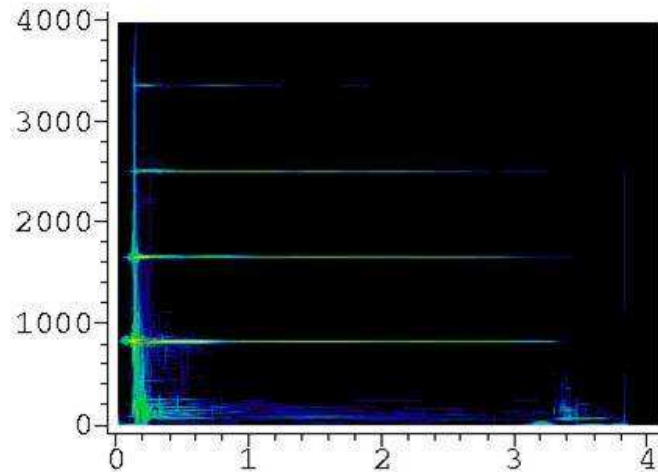


Figure 3.14: Time-frequency representation of a keystroke of piano, where typical audio structures are easily visible.

Another approach we investigated consists in exploiting the physical structures in electrocardiogram (ECG) signals, separating atrial and ventricular activity when atrial fibrillation occurs [125]. This case will be further illustrated in Section 6.6.2, but the reader can just observe Fig. 3.15 to have an idea of the two dominant structures present in this kind of ECG signals.

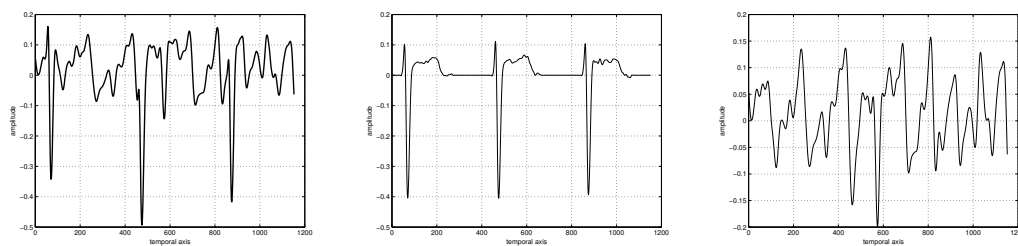


Figure 3.15: On the left, an ECG signal with atrial fibrillation. It contains two physically different structures, the ventricular activity (shown in the middle), and the atrial activity (on the right).

Notice that MCD is just a principle and does not tell anything about the kind of algorithm that should be used to analyze the information. In the previous pages we gave some hint about this problem for the case of natural images, but it will be tackled with much more details in the next three chapters.

Function Selection Methods

As seen in the previous chapter, we are interested in representing a signal using a multi-component dictionary that is built by the union of several sub-dictionaries. A MCD turns out to be overcomplete: this means it spans the signal space and its atoms form a linearly dependent set. Being the cardinality of the dictionary bigger than the dimension of the signal space, the decomposition of a signal is non-unique. This gives us the possibility of adaptation, choosing among many possible representations or approximations the one that (most) fits our purposes, even if this is not always a simple task. This chapter focuses on the characteristics we would like a signal decomposition to have and on how these can be obtained. We study algorithms properties and complexity, and we offer some examples of decompositions.

Remark that, during this dissertation, we make the difference between *exact representation* (otherwise simply wrote as representation) and *approximation*, depending on the fact that an error is tolerated or not. The latter problem has a extraordinary bigger practical impact, since, if the dictionary is complete, then every signal has a representation using all the atoms that it contains, but almost all signals require all the atoms to be exactly represented. To be more precise, the set of signals that have an exact representation without using all the atoms in the dictionary has Lebesgue measure zero in \mathbb{R}^n [176]. Nevertheless both problems are interesting and have been deeply studied.

4.1 Sparseness

Signal decompositions we are interested in have to be sparse, i.e. they should involve as few elements as possible. Intuitively, this request for sparseness can be interpreted as a need of compacting the information carried by a signal into the smallest amount of terms. If we are able to represent an image or any other information by using only few elements, it means that such elements can capture its main characteristics. We can say that they are meaningful because they do not only catch the most relevant parts of the information, but somehow they “understand” its structure.

Moreover, finding a sparse approximation finally means simplifying and the idea of replacing a complex object with a simpler (and possibly well organized) one has a strong impact on many applications. For example, a sparse approximation can be very helpful for compression, denoising, feature extraction, classification, source separation and many more signal analysis methods. Concerning the case of compression, however, the link between a sparse approximation and an efficient bit-rate is not straightforward and this problem will be practically tackled in Chapter 5.

Before going on with the discussion, let us just warn the reader that the concept of parsimonious representation that we address here as “sparseness” in the literature is sometimes also called “sparsity”. Such an ambiguity is also present in the English dictionary [5].

4.1.1 Mathematical Setting

This subsection contains some notations and definitions that will be used throughout the dissertation. We assume that the signal we are working with are in a finite-dimensional Hilbert space \mathcal{H} . If not differently stated we also fix $\mathcal{H} = \mathbb{R}^n$. The inner product is written as $\langle \cdot, \cdot \rangle$, and the corresponding Euclidean norm as $\| \cdot \|_2$.

The dictionary \mathcal{D} used to decompose signals is a finite collection of unit norm functions, also named atoms and labeled with g_i , where i belongs to the index set Ω . Of course, each atom is a signal itself, therefore belonging to \mathbb{R}^n . The size of the dictionary and consequently the cardinality of Ω are indicated with the letter $d = |\Omega| = |\mathcal{D}|$.

Since we want the dictionary to be both complete and redundant, we impose that \mathcal{D} spans the signal space and $d > n$. The redundancy factor of \mathcal{D} is defined as d/n . A dictionary can also be expressed using the corresponding synthesis matrix D , that is a matrix of size $d \times n$, whose columns are the atoms. The synthesis matrix maps every coefficient vector into a signal. Consequently the matrix D^T , where T denotes the transpose, is called the dictionary analysis matrix, and maps every signal into a coefficient vector.

Given a general real matrix M , we write its *Moore-Penrose generalized inverse* (or pseudoinverse) as M^+ [99]. The pseudoinverse satisfies the properties:

1. MM^+ and M^+M are symmetric.
2. $MM^+M = M$.
3. $M^+MM^+ = M^+$.

If M is square and non-singular, $M^+ = M^{-1}$.

A signal $f \in \mathbb{R}^n$ is represented as a linear combination of functions from \mathcal{D} :

$$f = \sum_{i \in \Omega} c_i \cdot g_i. \quad (4.1)$$

The coefficients form a vector that lies in \mathbb{R}^d and that will be called \mathbf{c} . Therefore, adopting a matrix notation, we can also write (see also Figure 4.1):

$$f = D\mathbf{c}. \quad (4.2)$$

Of course, in the case of approximation, the previous two expressions will be slightly modified in an unmistakable way. In this situation \hat{f} will be called approximant of f :

$$f \simeq \hat{f} = \sum_{i \in \Omega} \hat{c}_i \cdot g_i = D\hat{\mathbf{c}}. \quad (4.3)$$

Looking for a sparse signal decomposition means to use a cost function that counts the number of non-zero terms, i.e. the size of the support of a signal representation or approximation. This is done by the ℓ_0 quasi-norm:

$$\|\mathbf{c}\|_0 = |\text{support}(\mathbf{c})|. \quad (4.4)$$

In order to be a (vector) norm, a real-valued function $N(\mathbf{x})$ must satisfy the following four properties, where \mathbf{x}, \mathbf{y} are vectors and a is a scalar:

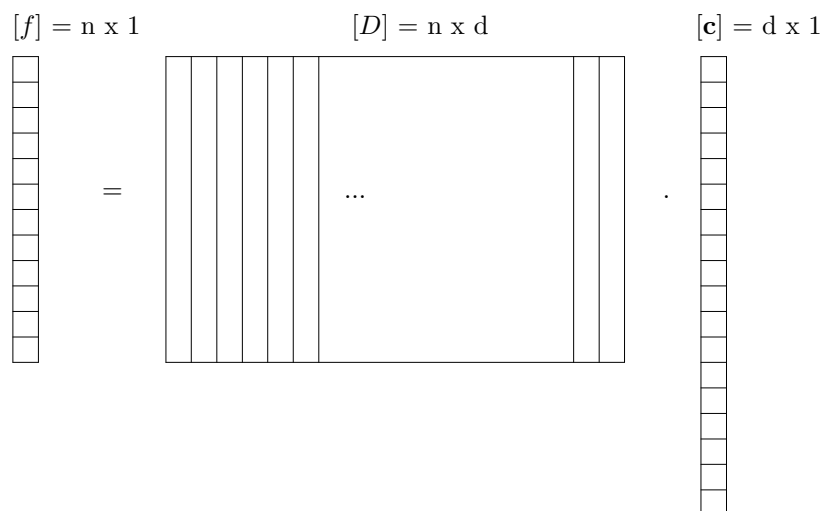


Figure 4.1: Matrix notation: f is a column-vector of size $n \times 1$. D has size $n \times d$ and each one of its columns corresponds to an atom of the dictionary. \mathbf{c} is the coefficient column-vector of size $d \times 1$. We ask \mathbf{c} to have as much zero-elements as possible.

1. $N(\mathbf{x}) \geq 0$.
2. $N(\mathbf{x}) = 0 \iff \mathbf{x} = 0$.
3. $N(\mathbf{x} + \mathbf{y}) \leq N(\mathbf{x}) + N(\mathbf{y})$.
4. $N(a\mathbf{x}) = |a| \cdot N(\mathbf{x})$.

These four axioms are familiar properties of Euclidean length in the plane. The ℓ_0 is a quasi-norm and not a norm since it does not respect property 4. In general, for any real number $p \geq 1$ we can define an ℓ_p -norm as:

$$\|\mathbf{c}\|_p = \left(\sum_{i \in \Omega} |c_i|^p \right)^{1/p}, \quad (4.5)$$

The same definition can be extended to any strictly positive, real number p , observing that if $p < 1$ property 3 is not respected. It can also be noticed that $\|\mathbf{c}\|_0 = \lim_{p \rightarrow 0} \|\mathbf{c}\|_p$. Moreover, the ℓ_∞ norm can be defined as:

$$\|\mathbf{c}\|_\infty = \sup_{i \in \Omega} |c_i|. \quad (4.6)$$

Finally, let us remark that the function in (4.5) is convex if and only if $p \geq 1$. This will have a dramatic impact in the following.

Operator Norms

Here we briefly introduce the norm of a matrix that we will use later on in this and next chapters. Considering a matrix M as an operator that maps two finite-dimensional vector spaces, we can define:

$$\|M\|_{p,q} = \max_{\mathbf{z} \neq 0} \frac{\|M\mathbf{z}\|_q}{\|\mathbf{z}\|_p} = \max_{\|\mathbf{z}\|_p=1} \|M\mathbf{z}\|_q. \quad (4.7)$$

This measure represents the factor by which the operator M changes the length of a vector. For the general properties of operators norm, we refer the reader to [99].

Dictionary Coherence Measures

We summarize some properties of a dictionary studying the interaction among atoms. The *coherence* μ of a dictionary equals the maximum inner product (in absolute value) between two distinct atoms (see [61]):

$$\mu = \max_{i \neq j} |\langle g_i, g_j \rangle|. \quad (4.8)$$

This quantity gives information about how much two atoms can look alike. Roughly speaking, a dictionary is said incoherent if μ is small. Of course, an orthonormal basis has coherence zero, while in [94] it is shown that a union of two orthonormal bases has coherence at least equal to $n^{-1/2}$.

Tropp in [174, 175] (but see also [59]) refines the idea of coherence, introducing a new parameter named *cumulative coherence*:

$$\mu_1(m) = \max_{|\Lambda|=m} \max_{i \notin \Lambda} \sum_{\lambda \in \Lambda} |\langle g_i, g_\lambda \rangle|, \quad (4.9)$$

where it is assumed that $\mu_1(0) = 0$. The cumulative coherence $\mu_1(m)$ measures how much a collection of m atoms can resemble a fixed distinct atom.

It is easy to see that $\mu_1(1) = \mu$ and $\mu_1(m) \leq \mu \cdot m$ [176]. So the cumulative coherence of an orthonormal basis is zero for every $m \geq 0$. It can also be observed that $\mu_1(m)$ increases monotonically with m :

$$\mu_1(m+1) - \mu_1(m) \geq 0. \quad (4.10)$$

Moreover, the cumulative coherence is a subadditive function.

4.1.2 Other Cost Functions

We have already said that among all the possible signal decompositions, we are looking for the sparsest one and for this purpose we use the ℓ_0 cost function. However, other cost functions are possible. Gribonval and Nielsen in [93] have studied a large class of admissible sparseness measures, showing how, in order to promote sparsity, a cost function should not charge for zero coefficients, while it should charge proportionally more for small coefficients than for big ones. Formally the class of sparseness measures is defined as the set of all non-decreasing functions $f : [0, \infty) \rightarrow [0, \infty)$, not identically zero, with $f(0) = 0$ and such that $t \rightarrow f(t)/t$ is non-increasing on $(0, \infty)$.

Without going into details, let us mention that the ℓ_0 quasi-norm and the ℓ_1 norm (also known as *sum norm*) both have these properties. On the other hand, the ℓ_2 norm (or *Euclidean norm*) does not meet all the requirements for being a cost function in the sense specified in [93]. This also explains the limits of the Method of Frames, as we will see in Section 4.7.1.

Figure 4.2 illustrates the concept of sparsity-preserving cost functions. Let us restrict to 2-D and suppose one wants to compute

$$\min_{b_1, b_2} |b_1|^p + |b_2|^p \quad \text{s.t.} \quad \|f - b_1 g_1 - b_2 g_2\|_2 \leq \epsilon, \quad (4.11)$$

where the constraint is represented in the figure by the dashed line (simplifying, in general it is not linear!). It is clear that if $p \leq 1$, then the coefficient involved in the decomposition is just one, while for bigger values of p , this property is lost.

Another important cost function is the Shannon entropy, roughly proportional to the number of bits necessary to represent the coefficient vector. Let us conclude this section highlighting that the ℓ_1 norm is the only cost function that is also convex. A simple proof can be found for example in [176].

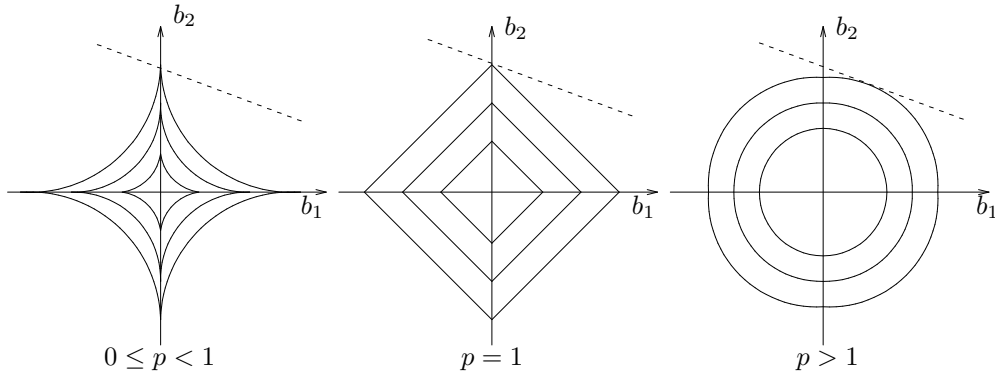


Figure 4.2: Capacity of ℓ_p cost functions to preserve sparseness.

4.2 Basic Original Problems

Up to now, dealing with the concept of sparseness, we met two similar classes of problems that are formally expressed in the following.

The first problem is the *sparsest signal exact representation*:

$$(P_0) \quad \min_{\mathbf{c} \in \mathbb{R}^d} \|\mathbf{c}\|_0 \quad \text{s.t.} \quad D\mathbf{c} = f. \quad (4.12)$$

The second one is the *error-constrained approximation* which, for a fixed tolerance for the error (ϵ), can be written as:

$$\min_{\mathbf{c} \in \mathbb{R}^d} \|\mathbf{c}\|_0 \quad \text{s.t.} \quad \|f - D\mathbf{c}\|_2 \leq \epsilon. \quad (4.13)$$

It is trivial to see how (P_0) arises from (4.13) just by setting the tolerance to zero.

The latter minimization knows multiple variations and it can be reformulated in different fashions. For example in approximation theory we can often find the following problem, called *m-term approximation* [47, 172]:

$$\min_{\mathbf{c} \in \mathbb{R}^d} \|f - D\mathbf{c}\|_2 \quad \text{s.t.} \quad \|\mathbf{c}\|_0 \leq m. \quad (4.14)$$

Another issue of (4.13) is given by the *subset selection problem*, often present in statistics:

$$(P_{2-0}) \quad \min_{\mathbf{c} \in \mathbb{R}^d} \|f - D\mathbf{c}\|_2^2 + \tau^2 \|\mathbf{c}\|_0. \quad (4.15)$$

In this last problem the cost function is a trade-off between the error, expressed by a classical Mean Square Error (MSE) and a penalty term that counts the number of elements involved in the approximation. When the parameter τ goes to zero, the approximation will involve more and more atoms until it will reach the exact representation of f . On the other hand, if we let τ grow up to $\|f\|_2$, the unique solution of (P_{2-0}) will be given by the zero vector.

Observe that all the previous four problems will select coefficient vectors that determine a dictionary subset composed by linearly independent functions. If this was not the case, some atoms could be discarded in order to diminish the value of the cost function.

Before facing the challenging question of how to solve these problems, let us spend few words justifying and explaining the notation that we adopted (and we will adopt in the following) to address them. We often label a problem as (P_{a-b}) , where a stands for the norm used for measuring the error, while b indicates the norm used to promote the sparsity of the coefficients. Of course, in the case of exact representation there is just one term in the cost function as it happens for (P_0) and this suggests the kind of norm used for the coefficient vector.

4.2.1 And Their Direct Solutions

Suppose one wants to find a solution to (P_0) . Unfortunately, as previously observed, the ℓ_0 quasi-norm is not a convex function and so, in a general situation, solving such a problem means testing all the representations of f over \mathcal{D} and selecting among them the one that involves the minimum number of terms. It therefore appears obvious how such a strategy is not feasible, even for relatively small values of n and d .

Concerning the approximation problems, Natarajan in [132] and Davis, Mallat and Avellaneda in [44] have showed how, in an unrestricted case, the minimization of Eq. (4.14) is NP-hard. This is also the case of the other two problems (4.13) and (4.15).

Therefore, the rest of the chapter presents several sub-optimal algorithms that have limited complexity, but in general provide sub-optimal solutions. However, we will see also that there exist particular situations where they are able to find the solution to the previous problems in polynomial time.

4.3 Thresholding the Projections

A very simple and naive method to select atoms that approximate f over \mathcal{D} is given by considering all the orthogonal projections of the signal on the atoms and selecting the biggest m in absolute value. This, in general, means finding the scalar products plus a (computationally negligible) sorting, i.e. the complexity is $O(n \cdot d)$. Moreover, if particular classes of orthonormal dictionaries ($d = n$) are adopted, the computational load can be strongly reduced, thanks to the use of filters to compute the coefficients. For example the complexity is $O(n \log_2 n)$ for a Discrete Cosine Transform, and $O(n)$ a Discrete Wavelet Transform [118, 179].

This procedure is a kind of nonlinear approximation [47, 172]. Sometimes one can be interested in substituting the sorting with a more simple procedure of shrinkage. Going more into details, let us consider two possible kinds of thresholding [62]: soft and hard, illustrated in Figure 4.3 and mathematically described as:

$$\begin{aligned}\Theta_{\eta}^{hard}(x) &= x \cdot \mathbf{1}_{(|x| > \eta)} \\ \Theta_{\eta}^{soft}(x) &= \text{sign}(x) \cdot (|x| - \eta)_+, \end{aligned} \quad (4.16)$$

where η is the threshold value, $(\cdot)_+$ stands for the positive part and $\mathbf{1}_{(p)}$ is 1 if p is true and 0 otherwise.

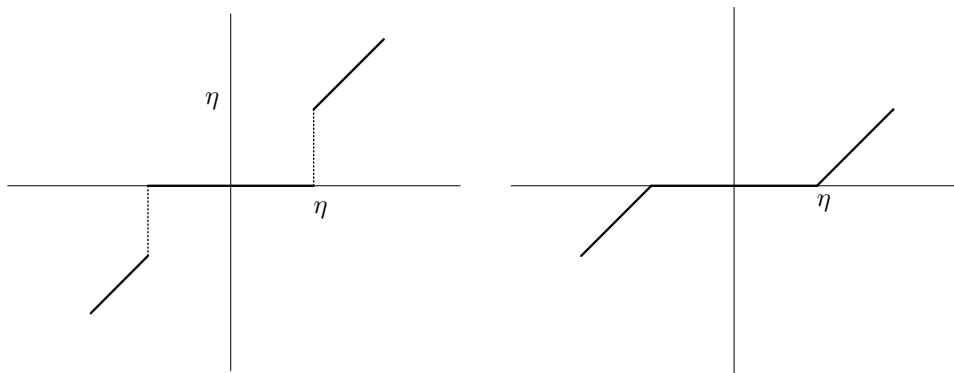


Figure 4.3: Hard and soft thresholding functions.

Performing a hard thresholding of the projections onto an orthonormal dictionary, one obtains the solution to problem (P_{2-0}) , while performing a soft thresholding one solves the problem:

$$\min_{\mathbf{c} \in \mathbb{R}^d} \|f - D\mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_1, \quad (4.17)$$

that will be studied in Section 4.5 (see [34, 62]). Note that for a orthonormal basis, both solutions have the same support and they differ only for the amplitude of the coefficients.

If we study the case where \mathcal{D} is redundant, but built by the union of several orthonormal bases, there exist very interesting results that show how it is still possible to recover the solution of Eq. (4.17) through the *Block Coordinate Relaxation* method [159, 160, 161]. For more details see Section 4.5.

When \mathcal{D} is a general overcomplete dictionary the situation becomes much more difficult. Also in this case thresholding can provide an m -term approximation of the signal, but there is no guarantee on the quality of such a method. A study of shrinkage operators for certain particular types of redundant dictionaries can be found in [21], where tight wavelet frames and Gabor Banach frames are considered. A work in progress by Gribonval et al. [97] provides a first sufficient condition for thresholding to be able to recover a sparse approximation of a noisy signal. This condition depends on the dictionary, the noise and the signal. Another very recent work which testifies that thresholding algorithms, even if very simple, are far from being unhelpful can be found in [65]. In this paper, the shrinkage of the coefficients of an overcomplete expansion is studied taking inspiration from [160, 161] and it is interpreted as a first iteration of an algorithm that solves the Basis Pursuit Denoising problem (see Section 4.5).

4.3.1 Example

Let us now provide an example of signal approximation by shrinkage. In particular, the procedure can be summarized into three steps:

Algorithm 4.1: Selection algorithm based on thresholding

Require: $\mathcal{D} = \{g_i\}_{i \in \Omega}$, define a threshold

- 1: PROJECT: Compute $\langle f, g_i \rangle, \forall i \in \Omega$.
 - 2: THRESHOLD: Threshold the projections. This specifies a sub-dictionary $\mathcal{D}_* \subset \mathcal{D}$.
 - 3: RE-PROJECT: Compute again the coefficients by re-projecting the signal onto the subspace spanned by the elements of the dictionary selected in Step 2.
-

Formally, being \mathcal{D}_* the sub-dictionary selected in Step 2, the new approximant f_* is found in Step 3 in the following way:

$$f_* = D_*(D_*)^+ f = D\mathbf{b}_*, \quad (4.18)$$

where \mathbf{b}_* contains the coefficients of the approximation.

The simple 1-D input signal of this example has a sinusoidal structure with two discontinuities (see the continuous line in Figure 4.5). Its length is 128 samples. The redundant dictionary used to decompose the signal is given by the union of two sub-dictionaries. A cosine packet (CP) with depth 3 and a wavelet *Symmlet-4* orthonormal basis [118]. Figure 4.4 shows the dictionary synthesis matrix D whose columns correspond to the atoms. The size of D is 128×640 , the first 512 columns are the CP atoms while the last 128 are the wavelet ones.

Figure 4.5 shows the input original signal (continuous line), together with the signals reconstructed selecting the biggest 15 (dashed line) and 23 coefficients (dotted line). The mean square errors are 0.080 for the approximation with 15 elements and 0.071 for the other one. Figure 4.6 represents

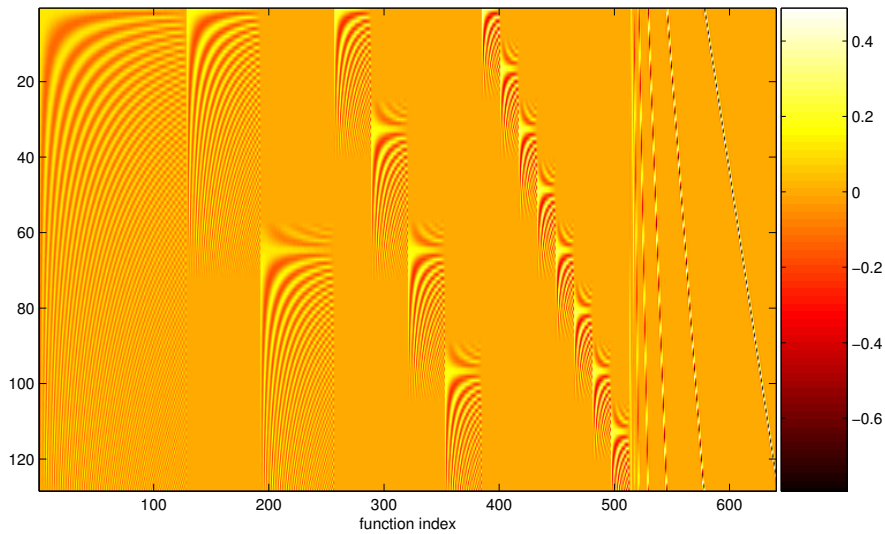


Figure 4.4: Representation of the redundant dictionary used in this example. Each column corresponds to an atom, the first 512 columns are the CP functions, the last 128 columns are wavelet functions.

the amplitudes of the coefficient vectors found with this method when m equals 15 (on the left) and 23 (on the right). The horizontal axis of this graphic gives the index of the basis functions in the dictionary, while the vertical axis specifies the amplitude of the coefficient of that function. In this way one can see what are the coefficients c_i of Eq. (4.3) and which atoms have been selected.

4.4 Greedy Algorithms

Greedy algorithms are known in statistics literature under the name of *forward selection*, *backward elimination*, *variable selection*, *stepwise regression*, etc. [124]. This class of algorithms was then introduced in the signal processing world by Mallat and Zhang in [120], where they presented Matching Pursuit (MP). Afterward, many variations of this algorithm were developed, among which the most important ones are Orthogonal Matching Pursuit (OMP) [144] and Weak Matching Pursuit (Weak-MP) [171].

Greedy algorithms iteratively build an approximant of a signal f by selecting the atom that maximizes a certain similarity measure with the residual part of the signal. After, such an atom is used to update the current approximation and the residual is computed again. Basically, fixing $r_0 = f$, each iteration $k : k \geq 0$ can be interpreted as composed by two steps:

1. A *selection* step where an atom $g_{i_k} \in \mathcal{D}$ is chosen, given r_k .
2. A *projection* step where an approximant $f_{k+1} \in \text{span}(\{g_{i_j}\}_{j=0\dots k})$ and a residual $r_{k+1} = f - f_{k+1}$ are generated.

The selection step at iteration k , can be generally formulated as the maximization of a similarity measure $C(r_k, g_i)$ between the signal to approximate (the residual at the k th iteration) and the

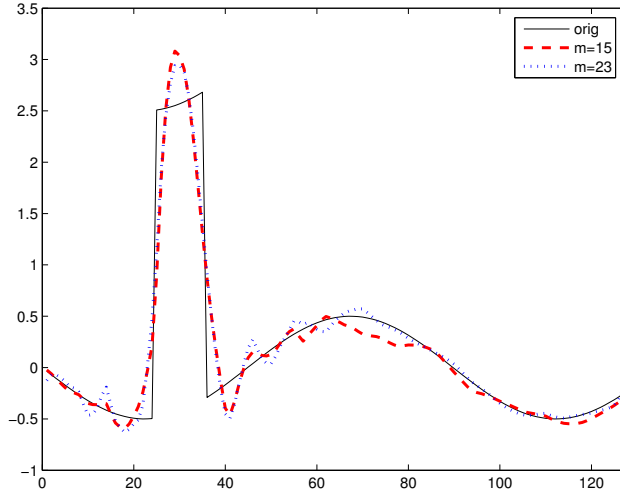


Figure 4.5: Original signal and reconstructions obtained by the thresholding method with 15 and 23 coefficients. The MSE obtained with 15 elements is 0.080, while with 23 is 0.071.

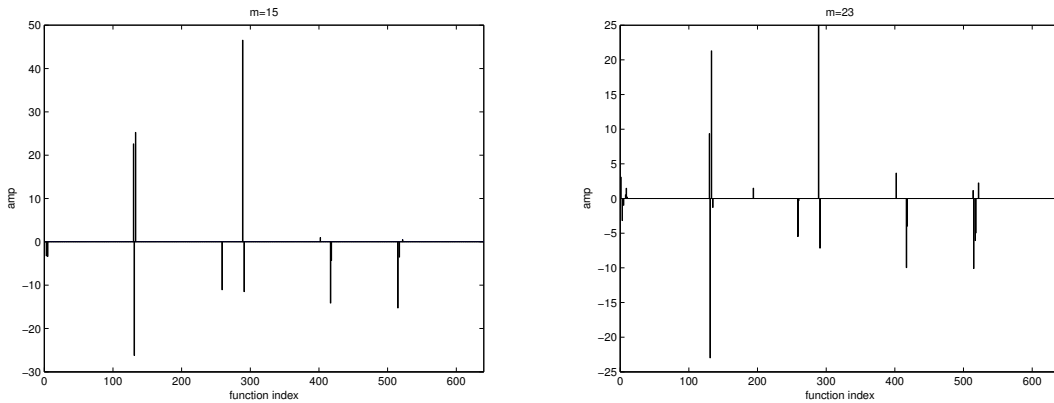


Figure 4.6: On the left: the amplitudes of the coefficients for the approximation with 15 elements. On the right: the amplitudes of the coefficients for the approximation with 23 elements.

dictionary atoms:

$$g_{i_k} = \arg \max_{g_i \in \mathcal{D}} C(r_k, g_i). \quad (4.19)$$

Matching Pursuit, also called *Pure Greedy Algorithm* in the approximation theory community (see [172]), uses the modulus of the scalar product as similarity measure, i.e. $C(r_k, g_i) = |\langle r_k, g_i \rangle|$. More generally, Weak-MP allows an additional flexibility factor $\alpha \in (0, 1]$ allowing the selected atom g_{i_k} to be such that $|\langle r_k, g_{i_k} \rangle| \geq \alpha \max_{i \in \Omega} |\langle r_k, g_i \rangle|$, where Ω is the set of indexes of the dictionary elements, as explained in Sec. 4.1.1. The sub-optimality factor α , does not prevent the greedy algorithm from converging to a solution (i.e. $\lim_{k \rightarrow \infty} \|r_k\|_2^2 = 0$, see [103]), though may affect negatively the speed of the convergence.

The projection step determines whether Matching Pursuit or Orthogonal Matching Pursuit is in use. The former just guarantees that the atom selected at iteration k is orthogonal to the residual r_{k+1} [120]. The latter constructs the approximant f_k by finding an orthogonal projection of f over the space spanned by all selected atoms until iteration k [144].

4.4.1 Matching Pursuit

Here we give a short description of the Matching Pursuit algorithm. For a more detailed explanation we refer to [120]. At the first iteration the signal f is decomposed as follows:

$$f = \langle g_{i_0}, f \rangle g_{i_0} + r_1, \quad (4.20)$$

where r_1 is the residual component after having approximated f in the direction of g_{i_0} . Since r_1 and g_{i_0} are orthogonal, it follows that

$$\|f\|_2^2 = |\langle g_{i_0}, f \rangle|^2 + \|r_1\|_2^2. \quad (4.21)$$

To minimize $\|r_1\|_2$, MP chooses g_{i_0} such that the absolute value of the projection $|\langle g_{i_0}, f \rangle|$ is maximal. Applying iteratively such a procedure, after m iterations we obtain:

$$f = \sum_{k=0}^{m-1} \langle g_{i_k}, r_k \rangle g_{i_k} + r_m, \quad (4.22)$$

where $r_0 = f$ and r_k is the residual after the k th step. In [118] it is proved that r_k converges to zero when k tends to infinity. The convergence is exponential in the case of finite dimensional signal spaces. Following (4.21), we can express the energy conservation of MP as:

$$\|f\|_2^2 = \sum_{k=0}^{m-1} |\langle g_{i_k}, r_k \rangle|^2 + \|r_m\|_2^2. \quad (4.23)$$

The convergence of MP depends on the structure of the dictionary, the search strategy and the signal f that has to be approximated. In [118] it is shown that, in finite dimension, there exist two real numbers $\alpha, \beta \in (0, 1]$ such that for all $k \geq 0$ the following relation holds:

$$\|r_{k+1}\|_2 \leq (1 - \alpha^2 \beta^2)^{1/2} \cdot \|r_k\|_2, \quad (4.24)$$

where α is the optimality factor related to the strategy adopted to select the best atom in the dictionary, while β depends on the dictionary, representing its ability to capture the features of the input function f (see [73]). Equation (4.24) gives a simple upper bound of the decay of the approximation error. In the case of infinite dimension the convergence is no longer exponential but it is still ensured [103].

The complexity of MP depends on the strategy for the implementation in use. In Chapter 5 we show how a signal of n samples can be decomposed with m atoms with a complexity of the order of

$$C \cdot m \cdot s \cdot n \log_2 n, \quad (4.25)$$

where the constant C depends on the strategy adopted for atom selection and s depends on the size of the dictionary. This is achieved for a particular class of dictionaries, by means of the Fast Fourier Transform (FFT). More details, will be given in Chapter 5.

Being MP iterative, a stopping criterion is needed. Of course the algorithm can be arrested when a maximum number of atoms has been selected, or when a target error is achieved. Another

stopping criterion may be found by looking at the amplitude of the projections and arresting the decomposition when they go under a certain threshold. A very interesting method to evaluate if the algorithm is able to further select a “good” atom at the next iteration can be found in [91]. However, from a computational point of view, this technique is far from being simple and it can never be applied in practical situations. For coding purposes much more sophisticated approaches are needed, namely based on rate-distortion techniques. We will present two of them in the next chapter in the case of still picture and video compression. Another example can be found in [136].

4.4.2 Recovery Conditions

In a famous example DeVore and Temlyakov have shown that MP is not necessary sparsity-preserving when the dictionary in use is not an orthogonal basis [48]. Suppose $\{\eta_i\}_{i=1}^{\infty}$ is an orthonormal basis in $L^2(\mathbb{R})$. Define the element

$$g = A\eta_1 + A\eta_2 + aA \sum_{i=3}^{\infty} (i(i+1))^{-1/2} \eta_i,$$

with $A = (33/89)^{1/2}$ and $a = (23/11)^{1/2}$, so that $\|g\|_2 = 1$. Define the dictionary $\mathcal{D} = \{g\} \cup \{\eta_i\}_{i=1}^{\infty}$. Let $f = \eta_1 + \eta_2$ be the target function and let f_m be the m -term approximant generated by MP. Ideally we should have $f_m = f$ for $m \geq 3$. In reality we get the much weaker result:

$$\|f - f_m\|_2 \geq m^{-1/2}, \text{ for } m > 3.$$

This situation is further analyzed in [172] and well explained by Chen in [33] where he adds the example illustrated in Figure 4.7. This consists in the decomposition of a signal composed by two sinusoids at two closely spaced frequencies over a redundant discrete cosine dictionary. The image on the bottom left of the figure shows how MP fails at recovering the two frequencies because the atom selected at the first iteration tries to capture both of them and all the further iterations are spent to correct this mistake. This can be seen as a lack of resolution in the MP decomposition. Motivated by this problem, a modification of MP was proposed, under the name of High Resolution Pursuit [102]. The right-hand side of Figure 4.7 illustrates the decompositions obtained by two other methods (Basis Pursuit and the Method of Frames) that will be introduced later on this chapter.

One can conclude that MP is myopic, i.e. it is not always able to see the real structure of the signal, and this mainly because of its greedy character. This can be interpreted also as a lack of global view.

However, there are plenty of cases where greedy algorithms do a good job even working with redundant dictionaries. Moreover it is clear how, in general, the possibility of greedy algorithms to recover “correct” atoms depends on the dictionary in use! Recently, new theoretical results have shown that it is possible to state hypotheses under which MP, OMP and Weak-MP are able to recover the optimal set of atoms in order to represent or approximate a signal. Such results give a more precise insight into the algorithm performances and, as previously said, they depend on the geometric properties of the dictionary.

The first results state sufficient condition for OMP to be able to recover the exact sparse representation of a signal. Suppose that the atoms that index the optimal solution to (P_0) are indexed in the set Γ , and \mathbf{c}_{Γ} is the optimal representation of f over \mathcal{D} . Therefore, $\Gamma = \text{support}(\mathbf{c}_{\Gamma})$. Let also D_{Γ} be the dictionary subset containing only the atoms indexed in Γ . Tropp in [174] has shown that, a sufficient condition for OMP to recover \mathbf{c}_{Γ} after $|\Gamma|$ steps is that:

$$1 - \sup_{g \notin D_{\Gamma}} \|D_{\Gamma}^+ g\|_1 > 0. \quad (4.26)$$

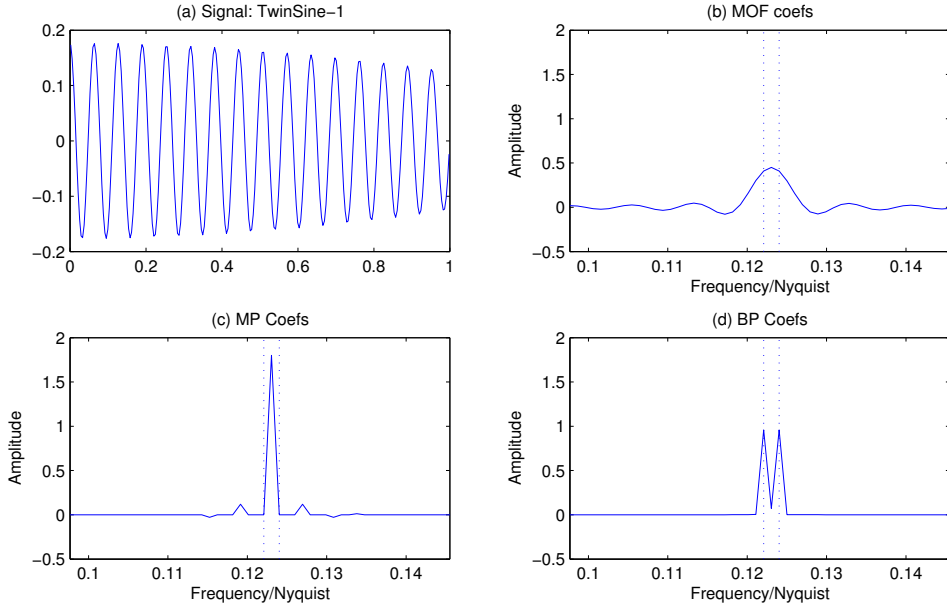


Figure 4.7: Decomposition of a double sinusoid signal over a redundant dictionary by the Method of Frames (MOF), MP and Basis Pursuit (BP). From [33].

The quantity on the left-hand side of Eq. (4.26) is called Exact Recovery Coefficient. Since we are unlikely to know the optimal atoms in Γ before starting a signal decomposition, the condition in (4.26) has practically a limited use. However, in order to obtain a condition that is easier to apply, one can establish a lower bound for the quantity that appears in Eq. (4.26). This is done in the following proposition appeared independently in [77] and [174].

Proposition 4.1 (Tropp [174] and Fuchs [77]) *Suppose that $\Gamma \subset \Omega$ is has cardinality smaller or equal to m . One can write that:*

$$1 - \sup_{g \notin D_\Gamma} \|D_\Gamma^+ g\|_1 \geq \frac{1 - \mu_1(m-1) - \mu_1(m)}{1 - \mu_1(m-1)}.$$

It follows that $\left(1 - \sup_{g \notin D_\Gamma} \|D_\Gamma^+ g\|_1\right) > 0$ whenever

$$\mu_1(m-1) + \mu_1(m) < 1.$$

Using Proposition 4.1 and Eq. (4.26) it is easy to prove that OMP solves (P_0) for every input signal that has an m -term representation over the dictionary whenever $\mu_1(m-1) + \mu_1(m) < 1$. This new result has the drawback of giving a more pessimistic bound than Eq. (4.26), but on the other hand it provides a condition that is possible to check before decomposing a signal. Therefore it has higher practical importance.

A more strict sufficient condition can be also expressed in terms of the dictionary coherence, just considering that $\mu_1(m) \leq m\mu$. In fact it can be proved that if $2m < (\mu^{-1} + 1)$, then OMP solves (P_0) for every input signal that has a m -term representation over \mathcal{D} . Moreover these representations are unique. The previous results for OMP can be extended to MP and Weak-MP (see [98]).

Sparse Approximation

Concerning the approximation case, one can establish some guaranties that greedy algorithms are able to recover the atoms of a m -term sparsest approximation of a signal f [98, 174]. Gribonval and Vandergheynst extended in [98] the results Tropp found for the particular case of OMP to the General MP (that is, MP, OMP, Weak-MP). The main achievements consist in the sufficient conditions that guarantee that General MP recovers the optimal set of atoms that generate the best m -term approximant f_m^{opt} . These are enunciated in Theorem 4.1. First of all, it is necessary that the optimal set Γ respects the Stability Condition [98]. If in addition some conditions are satisfied concerning the remaining residual energy at the k th iteration ($\|r_k\|_2^2$) and the optimal residual energy $\|r_m^{opt}\|_2^2$, then an additional atom belonging to Γ will be recovered.

Theorem 4.1 (Gribonval and Vandergheynst) *Let $\{r_k\}_{k \geq 0}$ be a sequence of residuals computed by General MP to approximate some $f \in \mathcal{H}$. For any integer m such that $\mu_1(m-1) + \mu_1(m) \leq 1$, let $f_m^{opt} = \sum_{\gamma \in \Gamma} c_\gamma g_\gamma$ be the best m -term approximation of f , and let $N_m = N_m(f)$ be the smallest integer such that*

$$\|r_{N_m}\|_2^2 \leq \|r_m^{opt}\|_2^2 \cdot \left(1 + \frac{m \cdot (1 - \mu_1(m-1))}{(1 - \mu_1(m-1) - \mu_1(m))^2} \right). \quad (4.27)$$

Then, for $1 \leq k < N_m$, General MP picks up a “correct” atom.

If no best m -term approximant exists, similar results are valid provided that one uses a modified version of Eq. (4.27) [98].

Moreover, a result establishes as well an upper bound on the decay of the residual energy in the approximation of a signal that depends on the internal coherence of \mathcal{D} , and a bound on how many “correct” iterations can be performed by the greedy algorithm depending on the dictionary and the energy of f_m^{opt} . Next result concerns the rate of convergence of the error energy, as well as the bound on how many “correct” iterations can be performed by the greedy algorithm.

Theorem 4.2 (Gribonval and Vandergheynst) *Let $\{r_k\}_{k \geq 0}$ be a sequence of residuals computed by General MP to approximate some $f \in \mathcal{H}$. For any integer m such that $\mu_1(m-1) + \mu_1(m) \leq 1$, let f_m^{opt} and $N_m = N_m(f)$ be defined as in Theorem 4.1. We have $N_1 \leq 1$, and for $m \geq 2$:*

- if $\|r_m^{opt}\|_2^2 \leq 3 \|r_1\|_2^2 / m$, then

$$2 \leq N_m < 2 + \frac{m}{1 - \mu_1(m-1)} \cdot \ln \frac{3 \cdot \|r_1\|_2^2}{m \cdot \|r_m\|_2^2} \quad (4.28)$$

- else $N_m \leq 1$.

4.4.3 Multiple Atoms MP

In [146, 147] we proposed a modified Matching Pursuit algorithm for which the constant C in Eq. (4.25) is much smaller than 1. At each iteration, $k_j - k_{j-1}$ atoms are selected and used to decompose the residual. Like in Eq. (4.22) we can write:

$$f = \sum_{j=0}^{J-1} \left(\sum_{k=k_j}^{k_{j+1}-1} \langle g_{i_k}, r_k \rangle g_{i_k} \right) + r_m, \quad (4.29)$$

with $k_0 = 0$ and $k_J = m$. At the k th iteration all the atoms of the dictionary are sorted according to the absolute values of the projection coefficients. Afterwards, starting from the one with highest

projection, all the atoms that are quasi-orthogonal are selected. We adopted this algorithm in order to obtain an important reduction in computational load. In fact selecting on average \bar{k}_j atoms at once it turns out that MP only needs m/\bar{k}_j iterations, reducing in this way the number of inner products which constitute the most computationally demanding part of the algorithm. At the k th iteration, the selected atom g_{k_j} has to be orthogonal to all the previous selected one at that iteration and this is achieved by working on the residual and considering the correlation between it and g_{k_j} . Moreover, of course, not all the quasi orthogonal are chosen. In details, one iteration of multiple MP can be described as follows:

Algorithm 4.2: Multiple Atoms MP Algorithm

Require: dictionary \mathcal{D} , residual signal r_k

- 1: Make a list of atoms according to their projections on r_k
 - 2: Select the first one and subtract it creating a temporal residual
 - 3: Check if the next atom has a projection on the temporal residual that is higher than a certain value (for example 95% of the one that he had originally). This corresponds to putting a threshold on the scalar product. If so, then subtract it and repeat, if not, skip to the next atom in the list
 - 4: There is a limit on how many atoms can be skipped, so it is avoided to select all the quasi-orthogonal atoms
-

Finally, there are two parameters to fix: the maximum number of atoms that can be skipped, and the percentage of projection required. The algorithm seems to be quite robust with respect to both of them. When adopting a dictionary composed by functions well localized in time/space like the one illustrated in Chapter 3, the condition of quasi-orthogonality is mainly verified when the atoms centers are distant.

Examples and performance of this modified Multiple Atoms MP concerning image approximation are shown in Chapter 5. The drawback of this method is that there is no more a guarantee that at each iteration the best atom will be selected as in the case of the full search MP. However, the resulting loss we experienced for both natural images and displaced frame differences is negligible [84, 146].

This method is similar to the fast MP implementation described by Mallat in [118]. When one controls the structure of the dictionary, one can update the projection of the atoms on the residual at the next step taking into account the correlation between atoms, according the following updating formula:

$$\langle g_i, r_{k+1} \rangle = \langle g_i, r_k \rangle - \langle g_{i_k}, g_i \rangle \langle g_{i_k}, r_k \rangle. \quad (4.30)$$

In particular the projections of the atoms orthogonal to the selected ones will not change.

4.4.4 Example

Let us now examine again the example given in Section 4.3.1, this time using greedy algorithms for decomposing the signal over the dictionary of Figure 4.4. Figure 4.8 shows the original input signal together with its approximants obtained by MP selecting 15 and 23 atoms. The errors are respectively 0.016 and 0.003. The selected atoms and their coefficients can be seen in Figure 4.9. These can be compared with the one selected by the other presented methods (see Figures 4.6, 4.11, 4.13, 4.15 and 4.18).

The approximants obtained by OMP are illustrated in Figure 4.10, while the amplitude of the coefficient vectors are in Figure 4.11. The error of the 15-term approximation is 0.012, while it is 0.003 for the 23-term approximation.

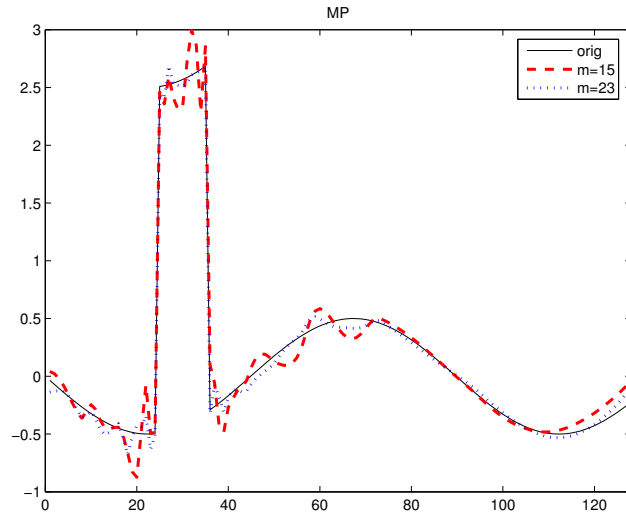


Figure 4.8: Original signal and reconstructions obtained by MP with 15 and 23 coefficients. The MSE are respectively 0.016 and 0.003.

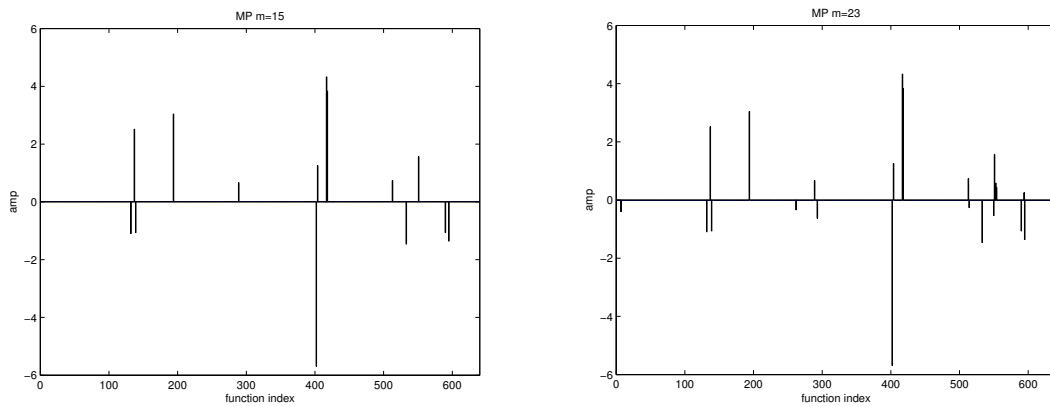


Figure 4.9: MP decomposition. On the left: the amplitudes of the coefficients for the approximation with 15 elements. On the right: the amplitudes of the coefficients for the approximation with 23 elements.

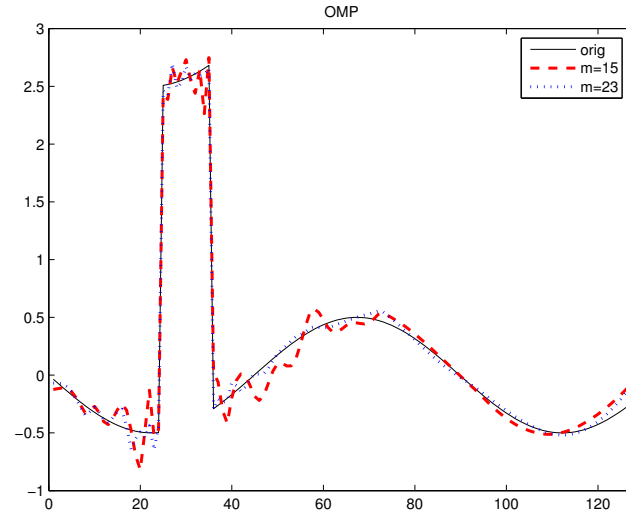


Figure 4.10: Original signal and reconstructions obtained by OMP with 15 and 23 coefficients. The MSE are respectively 0.012 and 0.003.

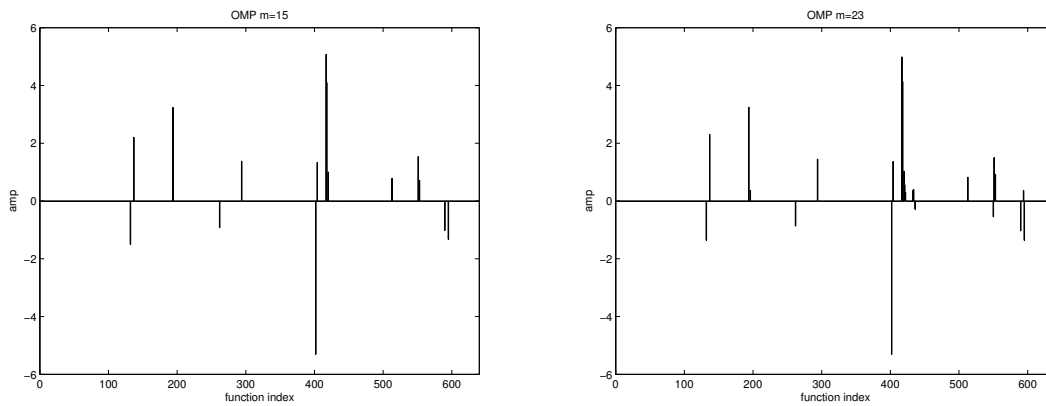


Figure 4.11: OMP decomposition. On the left: the amplitudes of the coefficients for the approximation with 15 elements. On the right: the amplitudes of the coefficients for the approximation with 23 elements.

4.4.5 Use and Variations of Greedy Algorithms

We want to conclude this section dedicated to greedy algorithms, stressing how, in the last years, this technique is becoming more and more employed. Undoubtedly, the most important example is the video codec proposed by Neff and Zakhor [15, 46, 133, 135, 136]. Furthermore, without the claim to be exhaustive, we can cite [84, 130, 184] again for what concerns motion compensated video coding, and [49, 53] as an example of non-canonical video compression, where the motion estimation part is left to the movement of the atoms through frames. We can also add references to [69, 75] for still images, and to [90, 108] for audio oriented applications, while in [128] MP is used for the analysis of multimodal signals. These examples offer an evidence of the increasing significance that MP is assuming in the scientific community.

Moreover many variations of greedy algorithms have been studied, mainly aimed at reducing the complexity. One, that we proposed and adopted for several applications, has just been illustrated in subsection 4.4.3. Another approach is the M-Term Pursuit [154], which relies on the concept of dictionary partitioning, i.e. splitting the dictionary into several (L) disjoint sub-dictionaries, each one carrying some specific information. Then, it iteratively finds an approximation, by selecting M atoms at a time, where $M \leq L$, followed by an orthogonal projection. The approximation performances of the M-Term Pursuit algorithm have been shown to contain the losses with respect to MP. Furthermore, it presents the advantage of a reduced computational complexity.

Another interesting method based on a similar idea of subdivision of the redundant dictionary is presented in [104], where the authors propose a structuring strategy that can be applied to any redundant set of functions, and which basically groups similar atoms together. A similarity measure based on coherence allows to represent a highly redundant sub-dictionary by a unique element. When the clustering is applied recursively, it naturally leads to the creation of a tree structure which can be used by a pursuit algorithm.

4.5 Convex Relaxation Methods

As we have seen above, (P_0) involves a non-convex minimization. Chen, Donoho and Saunders proposed a relaxation of such a problem, substituting the ℓ_0 quasi-norm with the convex ℓ_1 norm [33, 34]. This new problem was named Basis Pursuit (BP):

$$(P_1) \quad \min_{\mathbf{c} \in \mathbb{R}^d} \|\mathbf{c}\|_1 \quad \text{s.t.} \quad D\mathbf{c} = f. \quad (4.31)$$

Basis Pursuit is a principle, not an algorithm. However, it is possible to reformulate (P_1) as a Linear Programming (LP) problem (e.g. see [33]). We thus have at our disposal several techniques of low complexity to solve the Basis Pursuit problem (e.g. see [18, 34] and the web sites in [10, 11]).

4.5.1 Exact Recovery Conditions for Basis Pursuit

It is natural to study the relation between the solutions of problems (P_0) and (P_1) . This was firstly done by Donoho and Huo in [61], working with a union of two bases presenting extremely low coherence (namely, Fourier sinusoids and Dirac functions). Later, this result was generalized in [59, 66]. We now state the more recent results concerning general dictionaries. Suppose, as in Sec. 4.4.2, that the atoms that give the optimal solution of (P_0) are indexed in the set Γ , and \mathbf{c}_Γ is the optimal representation of f over \mathcal{D} . Therefore, $\Gamma = \text{support}(\mathbf{c}_\Gamma)$. Let also D_Γ be the synthesis matrix of the dictionary subset containing only the optimal atoms, i.e. the ones indexed in Γ . Tropp in [174] shows that (P_1) provides the unique optimal sparsest solution if the following condition is

respected:

$$1 - \sup_{g \notin D_\Gamma} \|D_\Gamma^\dagger g\|_1 > 0. \quad (4.32)$$

In this case BP recovers all the atoms of D_Γ and no others. Note that it is the same condition valid for greedy algorithms (see Eq. (4.26)). Thus, also in this case it is possible to state another stronger bound that has the advantage that it only depends on the dictionary and on m and therefore it can be computed before decomposing a signal. Using Proposition 4.1 and Eq. (4.32) we can state that BP (as well as OMP) is able to find the m -term sparsest representation if

$$\mu_1(m-1) + \mu_1(m) < 1. \quad (4.33)$$

There is another simpler but stronger sufficient condition based on the dictionary coherence. According to [59], BP will recover the optimal solution if $2m < (\mu^{-1} + 1)$. These results are valid for a general dictionary. If a dictionary has additional structure, one can prove less restrictive sufficient conditions. We refer the interested reader to [94, 176].

4.5.2 Approximation by Convex Relaxation Methods

As was done for the exact representation case, one can relax the sparsity constraint in the subset selection problem (P_{2-0}), ending up with a convex minimization:

$$(P_{2-1}) \quad \min_{\mathbf{b} \in \mathbb{R}^d} \frac{1}{2} \|f - D\mathbf{b}\|_2^2 + \gamma \|\mathbf{b}\|_1. \quad (4.34)$$

This problem is also known as Basis Pursuit Denoising (BPDN) [34]. It is possible to show that this minimization is equivalent to a Quadratic Programming (QP) problem and therefore we have at our disposal many techniques to solve it (e.g. see [17, 34] and the web sites in [10, 11]).

Note that if \mathcal{D} is orthonormal, the solution of (P_{2-1}) can be found by soft shrinkage of the coefficients [34, 62], while, if \mathcal{D} is a union of orthonormal bases, the problem can be solved using the Block Coordinate Relaxation method [159, 160, 161]. Basically this algorithm iteratively applies the soft shrinkage procedure to each sub-dictionary and, since usually it converges to a solution in few iterations, it turns out to be faster than QP.

4.5.3 A Bayesian Approach to Basis Pursuit Denoising

In this subsection the problem of signal approximation is studied from a Bayesian point of view. First, let us write the model of our data approximation, where \hat{f} is the approximant and r is the residual:

$$f = \hat{f} + r = D\mathbf{b} + r. \quad (4.35)$$

Assuming r to be an iid Gaussian set of variables, the probability that \hat{f} corresponds to f , given D and \mathbf{b} is:

$$p(f|D, \mathbf{b}) = \frac{1}{\sqrt{2\pi\sigma_r^2}} \cdot \exp\left(-\frac{\|f - D\mathbf{b}\|_2^2}{2\sigma_r^2}\right),$$

where σ_r^2 is the variance of the residual. In the approximation problem, one aims at maximizing the likelihood $p(\mathbf{b}|f, D)$. Formally, by the Bayes rule, we have

$$p(\mathbf{b}|f, D) = \frac{p(f|D, \mathbf{b}) \cdot p(\mathbf{b})}{p(f, D)},$$

and thus, being $p(f, D)$ fixed for a given signal and dictionary, it follows that the most probable signal representation is:

$$\mathbf{b}_P = \arg \max_{\mathbf{b}} p(f|D, \mathbf{b}) \cdot p(\mathbf{b}). \quad (4.36)$$

Let us now assume that the coefficients b_i are independent and have a Laplacian distribution with standard deviation σ_i :

$$p(b_i) = \frac{1}{\sqrt{2}\sigma_i} \cdot \exp\left(-\frac{\sqrt{2}|b_i|}{\sigma_i}\right).$$

From (4.36), by computing the logarithm, it follows that

$$\mathbf{b}_P = \arg \max_{\mathbf{b}} \left(\ln(p(f|D, \mathbf{b})) + \sum_i \ln p(b_i) \right) = \arg \min_{\mathbf{b}} \left(\frac{\|f - D\mathbf{b}\|_2^2}{2\sigma_r^2} + \sum_i \frac{\sqrt{2}|b_i|}{\sigma_i} \right). \quad (4.37)$$

Making the hypothesis that σ_i is constant for every index i , the previous equation means that the most probable \mathbf{b} is the one found by solving the BPDN problem [113].

Note that we have made two strong hypotheses: r is Gaussian, and b_i are Laplacian all with the same variance. We will see in the following of the dissertation two ways of partly removing these constraints.

4.5.4 Recovery Conditions for Basis Pursuit Denoising

It is now interesting to study the relation between the subset selection problem (P_{2-0}) and its convex relaxation (P_{2-1}). Pioneering works in this direction have been done by Tropp [174, 175], Fuchs [77], Gribonval and Nielsen [92, 95, 96], Donoho, Elad and Temlyakov [60]. Next theorem provides a sufficient condition for a coefficient vector which minimizes Eq. (4.34) to be supported inside the optimal set of indexes Γ .

Theorem 4.3 (Tropp) *Let us call f_m^{opt} the best approximant of f such that $f = D\mathbf{c}_\Gamma$, where Γ is the optimal index subset (in the sense of (P_{2-0})) and $|\Gamma| \leq m$. Suppose that the maximum inner product between the residual signal and any atom satisfies the condition*

$$\|D^*(f - f_m^{opt})\|_\infty < \gamma(1 - \sup_{i \notin \Gamma} \|D_\Gamma^+ g_i\|_1).$$

Then any coefficient vector \mathbf{b}_ that minimizes the function (P_{2-1}) must satisfy $\text{support}(\mathbf{b}_*) \subset \Gamma$.*

In particular, the following theoretical result shows how the trade-off parameters τ and γ are related.

Theorem 4.4 (Tropp) *Suppose that the coefficient vector \mathbf{b}_* minimizes the function (P_{2-1}) with threshold $\gamma = \tau/(1 - \sup_{i \notin \Gamma} \|D_\Gamma^+ g_i\|_1)$. Then we have that:*

1. *The relaxation never selects a non optimal atom since $\text{support}(\mathbf{b}_*) \subset \Gamma$.*
2. *The solution of the convex relaxation is unique.*
3. *The following upper bound is valid:*

$$\|\mathbf{c}_\Gamma - \mathbf{b}_*\|_\infty \leq \frac{\tau \cdot \left\| (D_\Gamma^* D_\Gamma)^{-1} \right\|_{\infty, \infty}}{1 - \sup_{i \notin \Gamma} \|D_\Gamma^+ g_i\|_1}. \quad (4.38)$$

4. The support of \mathbf{b}_* contains every index j for which

$$|\mathbf{c}_\Gamma(j)| > \frac{\tau \cdot \left\| (D_\Gamma^* D_\Gamma)^{-1} \right\|_{\infty, \infty}}{1 - \sup_{i \notin \Gamma} \|D_\Gamma^\dagger g_i\|_1}. \quad (4.39)$$

Note that, if the dictionary we are working with is orthonormal it follows that

$$\sup_{i \notin \Gamma} \|D_\Gamma^\dagger g_i\|_1 = 0 \text{ and } \left\| (D_\Gamma^* D_\Gamma)^{-1} \right\|_{\infty, \infty} = 1$$

and the previous theorem becomes much stronger. In particular we obtain that $\|\mathbf{c}_\Gamma - \mathbf{b}_*\|_\infty \leq \tau$ and $|\mathbf{c}_\Gamma(j)| > \tau$ [62, 175].

These results provide sufficient conditions for BPDN to recover the sparsest approximation. In such cases one can just use QP methods to solve (P_{2-1}) and be sure that the coefficient vector \mathbf{b}_* will contain all the atoms of the vector minimizing (P_{2-0}) . Unfortunately, the hypotheses of Theorems 4.3 and 4.4 in general cannot be checked since they depend on the optimal set Γ that, of course, is not known in advance. In order to solve this problem, Tropp developed in [175] a version of Theorem 4.4 based on the cumulative coherence function. Here, the hypotheses depend only on \mathcal{D} and on m that is the size of the optimal index set Γ .

Corollary 4.1 (Tropp) *Suppose that the vector \mathbf{b}_* solves (P_{2-1}) with threshold*

$$\gamma = \frac{1 - \mu_1(m-1)}{1 - \mu_1(m) - \mu_1(m-1)} \tau.$$

Then $\text{support}(\mathbf{b}_) \subset \Gamma$ and*

$$\|\mathbf{b}_* - \mathbf{c}_\Gamma\|_\infty \leq \frac{\tau}{1 - \mu_1(m) - \mu_1(m-1)}.$$

This result is mainly an application of Proposition 4.1 to Theorem 4.4.

Basis Pursuit and Basis Pursuit Denoising solve a global problem, considering all the signal as a whole. For this reason they are not affected by the “myopia” of greedy algorithms. This can be seen in the example of Figure 4.7, where BP is able to resolve the two frequencies correctly. This phenomenon is also known as *super resolution*. However, the recovery condition obtained for relaxed methods do not differ from the ones proved for greedy methods. Moreover in many applications the approximations found by relaxed algorithms do not outperform the ones given by greedy algorithms.

4.5.5 Example

We are going to give an example of the BPDN decomposition using again the signal and the redundant dictionary introduced in Section 4.3.1. The approximation \mathbf{b} found by solving (P_{2-1}) may present some components with negligible values due to the numerical computation. Therefore, a hard thresholding is performed in order to get rid of this insignificant elements. In this way, it is possible to measure the ℓ_0 quasi-norm of the thresholded coefficient vector which will be called \mathbf{b}_* . Finally, the approximant is computed as $f_* = D\mathbf{b}_*$. Of course this procedure is sensitive to the threshold adopted and one has to be careful not to choose a too high value.

Figure 4.12 shows the original signal along with two approximants obtained by BPDN. The dashed line refers to a 15-term approximation, obtained using in (P_{2-1}) a trade-off parameter $\gamma = 3.1$. The mean square error is 0.448. The dotted line refers to a 23-term approximation, obtained with $\gamma = 1.1$. The error is 0.107. The selected atoms and their coefficients can be seen in

Figure 4.13. All these results refer to the coefficient vectors \mathbf{b}_* computed with a threshold of 10^{-9} . Such value allows to count the number of non-zero coefficients but does not affect the quality of the representation. The MSE of f_* and the one of the reconstruction using the approximation vector found by BPDN without any thresholding differ only by a factor of 10^{-12} !

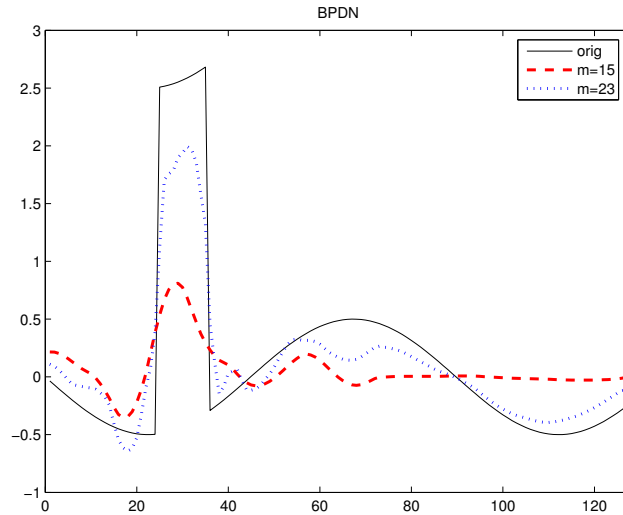


Figure 4.12: Original signal and reconstructions obtained by BPDN with 15 and 23 coefficients. The MSE are respectively 0.448 and 0.107.

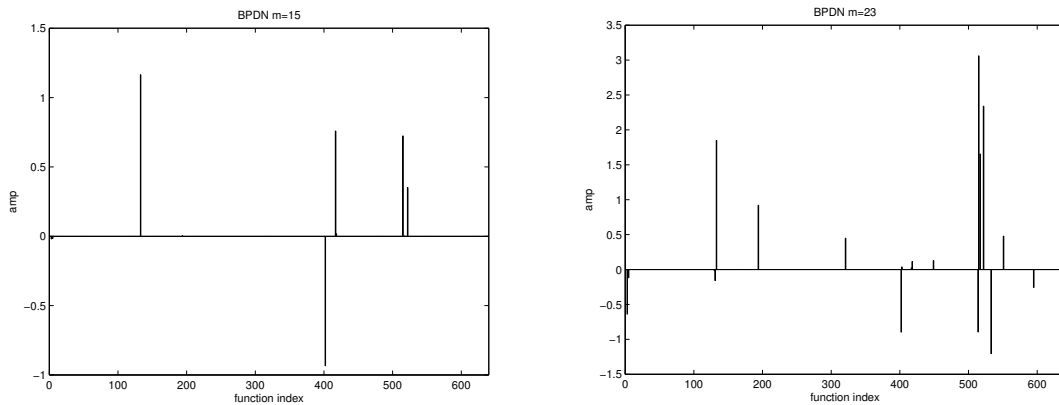


Figure 4.13: BPDN decomposition. On the left: the amplitudes of the coefficients for the approximation with 15 elements. On the right: the amplitudes of the coefficients for the approximation with 23 elements.

Projection

As just explained, the coefficient vectors found by numerically solving (P_{2-1}) are thresholded removing the negligible components, and in this way we are able to select a sparse support and thus

a subset of the dictionary. Let us name \mathcal{D}_* such sub-dictionary composed by the atoms corresponding to the non-zero elements of \mathbf{b}_* . Given this, there are no guarantees that the coefficients that represent f are optimal. The best result one can obtain is a bound on the maximum distance they can have with respect to the optimal approximation \mathbf{c}_Γ , as can be seen in point 3 of Theorem 4.4. It is however possible to overcome this limit by simply projecting f onto the subspace generated by \mathcal{D}_* . Doing so, a new approximation with the same support as the previous one is computed: \mathbf{b}_{**} . Formally the approximant found by BPDN after the projection step is given by:

$$f_{**} = D_* D_*^+ f = D \mathbf{b}_{**}, \quad (4.40)$$

where D_* is of course the synthesis matrix corresponding to the sub-dictionary \mathcal{D}_* . This orthogonal projection usually allows a quite significant improvement on the BPDN decomposition (e.g. see [52]). This is also the case in our example, where computing again the approximants by means of (4.40) leads to the results of Figure 4.14. The MSE are 0.078 for $m = 15$ and 0.009 for $m = 23$. Figure 4.15 shows the amplitudes of \mathbf{b}_{**} for the 15-term approximation (on the left) and the 23-term approximation (on the right), computed respectively with $\gamma = 3.1$ and 1.1, as well as the ones of Figure 4.13. Note that the supports of the coefficient vectors shown in Figure 4.13 and 4.15 are the same (respectively for $m = 15$ and $m = 23$). However this is not clear from the pictures because of the very small amplitude of certain coefficients in Fig. 4.13.

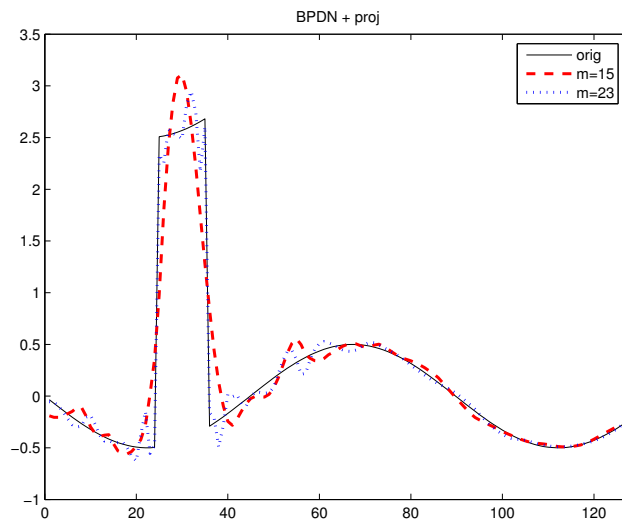


Figure 4.14: Original signal and reconstructions obtained with 15 and 23 coefficients by a BPDN decomposition followed by a projection step. The MSE are respectively 0.078 and 0.009.

4.6 Relaxed Approximation Using an L_1 Data-Fidelity Term

In this section we propose a slightly different problem we introduced in [86], where an L_1 data-fidelity term substitutes for the classical L_2 measure of the error:

$$(P_{1-1}) \quad \min_{\mathbf{b} \in \mathbb{R}^d} \|f - D\mathbf{b}\|_1 + \gamma \|\mathbf{b}\|_1. \quad (4.41)$$

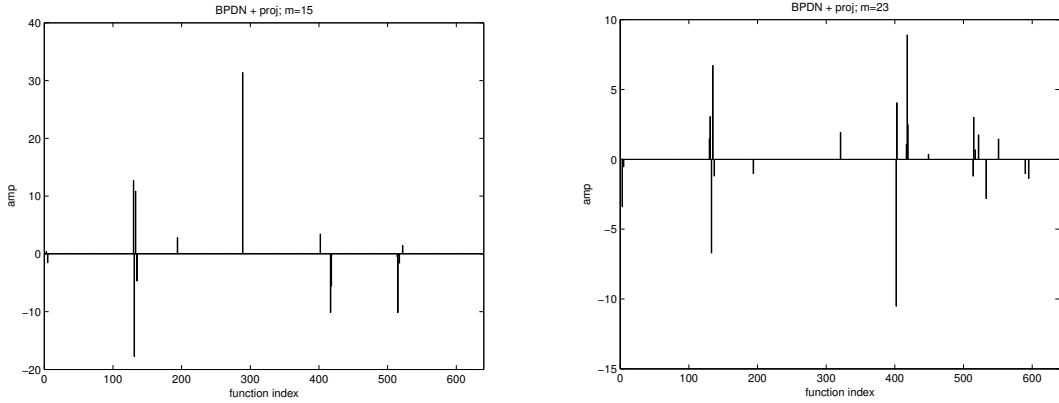


Figure 4.15: BPDN decomposition followed by a projection step. On the left: the amplitudes of the coefficients for the approximation with 15 elements. On the right: the amplitudes of the coefficients for the approximation with 23 elements.

In this way less importance is given to outliers, or “wild” signal samples, while keeping the same penalization for the sparseness of the coefficient vector.

In Section 4.5.3 we have seen a Bayesian formulation of the BPDN paradigm. We showed that, assuming the residual r to be an iid Gaussian set of variables, if the standard deviations of the Laplacian distribution of b_i are constant for every index i , the most probable signal approximation is the one found by BPDN. The assumption made about the Gaussianity of the residual is quite restrictive. For another particular problem, one could make the hypothesis that this residual has a Laplacian distribution. It is then possible to prove that the most probable signal representation can be found substituting the L_2 measure of the error with the L_1 . This leads to the problem (P_{1-1}) .

Recently the total variation based image denoising model of Rudin, Osher, and Fatemi (see [157]) has been modified by using the L_1 norm to calculate the fidelity term in the cost function [32, 137]. This modification brings new interesting implications, as can be seen for example in the works of Nikolova [138]. In [76] the problem of image restoration is considered, where an original scene f has to be recovered given its observation \hat{f} . The problem can be written as:

$$\hat{f} = Hf + w, \quad (4.42)$$

where H is a blurring matrix and w the additive noise. The authors of [76] propose to solve this problem minimizing the following cost function:

$$\min_f \|\hat{f} - Hf\|_1 + \gamma \|Rf\|_1, \quad (4.43)$$

where R is a regularization operator, usually a difference operator. Note the similarity between Eq. (4.43) and (P_{1-1}) . Our choice to introduce (P_{1-1}) from (P_{2-1}) , follows a similar idea, even if the background of the two problems is different.

The measure of the approximation error with ℓ_1 norm has been also used by Candes and Tao in [28, 29]. Moreover, in the Discussion of [175] Tropp imagines the situation where the ℓ_2 norm is not the most appropriate way to measure the error in approximating the input signal, but without giving further details.

It is important to observe that the minimization of (P_{1-1}) can be written as a Linear Program-

ming problem of the following form:

$$\min_{\mathbf{x}} \mathbf{v}^T \mathbf{x} \quad \text{s.t.} \quad A\mathbf{x} = f \quad \text{and} \quad \mathbf{x} \geq 0, \quad (4.44)$$

where \mathbf{v} is a vector of known coefficients. In order to show this equivalence (see also [159]) one should create a vector $\mathbf{u} = (\mathbf{u}_+, \mathbf{u}_-)$ with $\mathbf{u}_+, \mathbf{u}_- \geq 0$ such that $\mathbf{b} = \mathbf{u}_+ - \mathbf{u}_-$. The vector \mathbf{u}_+ contains only the positive components of \mathbf{b} , while the negative ones are in \mathbf{u}_- , but with a positive sign. In this way one can see that $\|\mathbf{b}\|_1 = \mathbf{1}^T \mathbf{u}$, where $\mathbf{1}$ is a column-vector of ones. The same can be done defining a vector $\mathbf{r} = (\mathbf{r}_+, \mathbf{r}_-)$, with $\mathbf{r}_+, \mathbf{r}_- \geq 0$ and

$$\mathbf{r}_+ - \mathbf{r}_- = f - (D, -D) \cdot \mathbf{u}.$$

It is now clear that Eq. (4.41) can be written as

$$\min_{\mathbf{u}, \mathbf{r}} \mathbf{1}^T \mathbf{r} + \gamma \mathbf{1}^T \mathbf{u} \quad \text{s.t.} \quad A \cdot (\mathbf{r}, \mathbf{u}) = f \quad \text{and} \quad \mathbf{u}, \mathbf{r} \geq 0,$$

with $A = (I, -I, D, -D)$, where I is a $n \times n$ identity matrix. Here we find the form of (4.44), with $\mathbf{v} = (\mathbf{1}, \gamma \mathbf{1})$ and $\mathbf{x} = (\mathbf{r}, \mathbf{u})$.

4.6.1 An Application to Signal Denoising

In this section we offer an example of the use of the proposed minimization problem, in the framework of signal denoising. Exactly as it was done for BPDN, the approximation found by solving (P_{1-1}) is thresholded, removing the numerically negligible components. In this way we are able to individuate a vector \mathbf{b}_* with a sparse support and thus a subset of the dictionary. Let us label \mathcal{D}_* the sub-dictionary composed by the all atoms corresponding to the non-zero elements of \mathbf{b}_* . There are no guarantees that the coefficients found by solving (P_{1-1}) that represent f over \mathcal{D}_* are optimal. These are, thus, recomputed projecting the signal onto the subspace spanned by the elements of \mathcal{D}_* and a new approximation \mathbf{b}_{**} is found. Of course, $\text{support}(\mathbf{b}_*) = \text{support}(\mathbf{b}_{**})$. Formally the approximant found after the projection step is:

$$f_{**} = D_*(D_*)^+ f = D\mathbf{b}_{**}. \quad (4.45)$$

Thus, the minimization of Eq. (4.41) is used only to select the dictionary subset. Note that this equation is the same as Eq. (4.40). The difference is that here \mathbf{b}_* is found by thresholding the solution of (P_{1-1}) , while in (4.40) it is found by thresholding the solution of (P_{2-1}) .

We now decompose a piecewise smooth signal affected by impulse noise. The dictionary used has redundancy factor 2 and is composed by the union of a wavelet *Symmetlet-4* orthonormal basis [118] and the respective family of footprints for all the possible translations of the Heaviside function (see [64]). At the beginning of this Section we have given a constructive proof of how (P_{1-1}) can be written as a Linear Programming. The results presented here as well as in Section 4.6.3 have been obtained using a LP technique based on the interior-point method [185]. Figure 4.16 shows the original noisy signal, and two reconstructions obtained by solving (P_{1-1}) on the left and (P_{2-1}) on the right, and then recomputing the coefficients by orthogonal projection as in (4.45). It can be seen how (P_{1-1}) is less sensible to wild samples given by the impulse noise, thanks to the ℓ_1 error penalization that allows the algorithm to select a better subset of functions. The MSE is 0.37 and 0.61 for (P_{1-1}) and (P_{2-1}) respectively, and remark that the MSE is not an error measure favorable to (P_{1-1}) since it is based on the Euclidean norm.

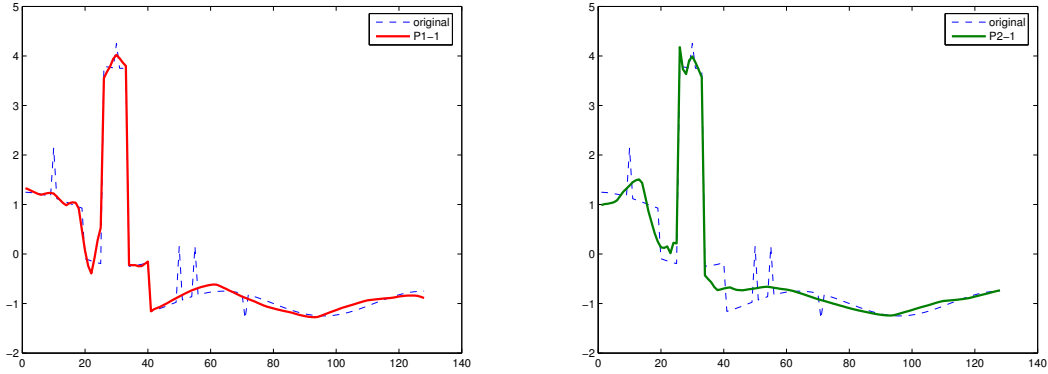


Figure 4.16: The original, noisy signal and the approximations obtained with 9 coefficients by solving (P_{1-1}) on the left and (P_{2-1}) on the right

4.6.2 Special Recovery Conditions

Let us now study the relationship between (P_{1-1}) and the following non relaxed minimization problem, where the error is still measured with the ℓ_1 norm, but the sparsity penalty factor is based on the ℓ_0 norm:

$$(P_{1-0}) \quad \min_{\mathbf{c} \in \mathbb{R}^d} \|f - D\mathbf{c}\|_1 + \tau^2 \|\mathbf{c}\|_0. \quad (4.46)$$

The cost function of this problem is a trade-off between the sparseness of the approximation and its distance from the input signal. Again (P_{1-0}) is not convex and here we wonder when and how solving (P_{1-1}) can help us in finding the solution of (4.46). An answer is provided by next theorem.

Theorem 4.5 *Let \mathbf{b}_* be the coefficient vector that minimizes (P_{1-1}) and let $\Gamma \subset \Omega$ be the optimal function subset found by solving the non-convex problem (P_{1-0}) . \mathcal{D}_Γ will be the sub-dictionary containing only the functions indexed in Γ . Suppose that $\sup_{i \notin \Gamma} \|D_\Gamma^+ g_i\|_1 < 1$, then we can state that if*

$$\gamma > \frac{\sqrt{n}}{1 - \sup_{i \notin \Gamma} \|D_\Gamma^+ g_i\|_1} \quad (4.47)$$

then $\text{support}(\mathbf{b}_*) \subset \Gamma$.

Proof: This proof is inspired by the proof of the Correlation Condition Lemma that appears in [175]. Let us call $\mathcal{D}_{\bar{\Gamma}}$ the complementary of \mathcal{D}_Γ on \mathcal{D} , such that $\mathcal{D} = \mathcal{D}_\Gamma \cup \mathcal{D}_{\bar{\Gamma}}$. Suppose that \mathbf{b}_* contains (at least) one element out of Γ , so we can write the cost function of (P_{1-1}) for both \mathbf{b}_* and its projection onto \mathcal{D}_Γ , that is $D_\Gamma^+ D\mathbf{b}_*$. Since \mathbf{b}_* minimizes (P_{1-1}) , we have:

$$\gamma (\|\mathbf{b}_*\|_1 - \|D_\Gamma^+ D\mathbf{b}_*\|_1) \leq \|f - DD_\Gamma^+ D\mathbf{b}_*\|_1 - \|f - D\mathbf{b}_*\|_1. \quad (4.48)$$

Let us now split the coefficient vector into two parts: $\mathbf{b}_* = \mathbf{b}_\Gamma + \mathbf{b}_{\bar{\Gamma}}$, where the former vector contains the components with indexes in Γ , while the latter the remaining components from $\bar{\Gamma} = \Omega \setminus \Gamma$. The left-hand term of (4.48) can be bounded as in [175] obtaining:

$$\gamma \left(\left(1 - \sup_{i \notin \Gamma} \|D_\Gamma^+ g_i\|_1\right) \cdot \|\mathbf{b}_{\bar{\Gamma}}\|_1 \right) \leq \gamma (\|\mathbf{b}_*\|_1 - \|D_\Gamma^+ D\mathbf{b}_*\|_1). \quad (4.49)$$

We now work with the right-hand side of (4.48):

$$\begin{aligned} \|f - DD_{\Gamma}^{\dagger} D\mathbf{b}_*\|_1 - \|f - D\mathbf{b}_*\|_1 &\leq \|D\mathbf{b}_* - P_{\Gamma} D\mathbf{b}_*\|_1 = \\ &\|(I - P_{\Gamma})D\mathbf{b}_*\|_1 \leq \|(I - P_{\Gamma})D\|_{1,1} \cdot \|\mathbf{b}_{\Gamma}\|_1, \end{aligned} \quad (4.50)$$

where $P_{\Gamma} = DD_{\Gamma}^{\dagger} = D_{\Gamma}D_{\Gamma}^{\dagger}$ is an orthogonal projector. Using this result together with (4.48) and (4.49) we obtain:

$$\gamma(1 - \sup_{i \notin \Gamma} \|D_{\Gamma}^{\dagger} g_i\|_1) \leq \|(I - P_{\Gamma})D\|_{1,1}. \quad (4.51)$$

The right-hand side of the previous equation is the maximum ℓ_1 norm of the columns of $(I - P_{\Gamma})D$, that is:

$$\|(I - P_{\Gamma})D\|_{1,1} = \max_{g \in D_{\Gamma}} \|g - P_{\Gamma}g\|_1 \leq \max_{g \in D_{\Gamma}} \|g - P_{\Gamma}g\|_2 \cdot \sqrt{n} \leq \max_{g \in D_{\Gamma}} \|g\|_2 \cdot \sqrt{n} = \sqrt{n}. \quad (4.52)$$

Finally, we have

$$\gamma(1 - \sup_{i \notin \Gamma} \|D_{\Gamma}^{\dagger} g_i\|_1) \leq \sqrt{n}. \quad (4.53)$$

If this inequality fails, then \mathbf{b}_* is supported in Γ . ■

Unfortunately, since the optimal set of functions is not known, this condition can not be tested before decomposing a signal. This is exactly the same situation found for the recovery conditions of BP, BPDN and greedy algorithms. However, the following corollary helps us, finding an additional condition based on the cumulative coherence $\mu_1(m)$.

Corollary 4.2 *If $|\Gamma| \leq m$ and $\mu_1(m-1) + \mu_1(m) < 1$, then $\text{support}(\mathbf{b}_*) \subset \Gamma$ if*

$$\gamma = \frac{\sqrt{n}(1 - \mu_1(m-1))}{1 - \mu_1(m-1) - \mu_1(m)}. \quad (4.54)$$

Proof: Equation (4.54) can be simply obtained by applying Proposition 4.1 to the result of Theorem 4.5. ■

The new sufficient condition of Eq. (4.54), even if more pessimistic than (4.47), can be numerically checked. However, also computing $\mu_1(m)$ for m and \mathcal{D} not too small can be quite computationally demanding.

4.6.3 Example

We conclude this section by facing the signal decomposition we previously solved using the Thresholding method, MP, OMP and BPDN. As well as done for BPDN and explained in Section 4.6.1, we solve (P_{1-1}) in order to select a dictionary subset with the help of a thresholding. At this point the approximants are computed following Eq. (4.45).

Figure 4.17 shows the original signal along with its approximants obtained by selecting 15 and 23 atoms. The errors are respectively 0.170 and 0.077. The approximation coefficients \mathbf{b}_{**} can be seen in Figure 4.18. The trade-off parameter γ equals 11.6 for the 15-term approximation and 11.22 for the other one, while the value used for thresholding the coefficients obtained by solving (P_{1-1}) is 10^{-9} . Obviously, the approximation obtained by this method is worse than many of the previous ones. However, as shown above, such an approach can be useful for particular situations like denoising.

Table 4.1 summarizes all the results obtained in this section when approximating the signal illustrated in Sec. 4.3.1. These results have no general relevance, since they refer to a particular

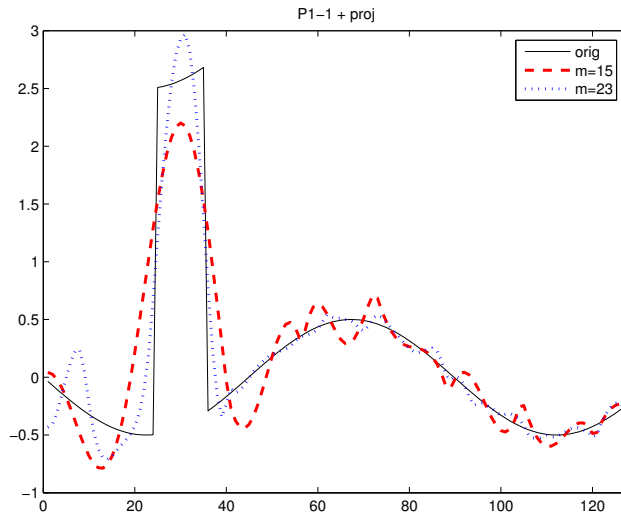


Figure 4.17: Original signal and reconstructions obtained with 15 and 23 coefficients. Approximations are found by defining a sub-dictionary solving (P_{1-1}) and then projecting the signal onto the subspace spanned by the atoms in the sub-dictionary (see Eq. (4.45)). The MSE are respectively 0.170 and 0.077.

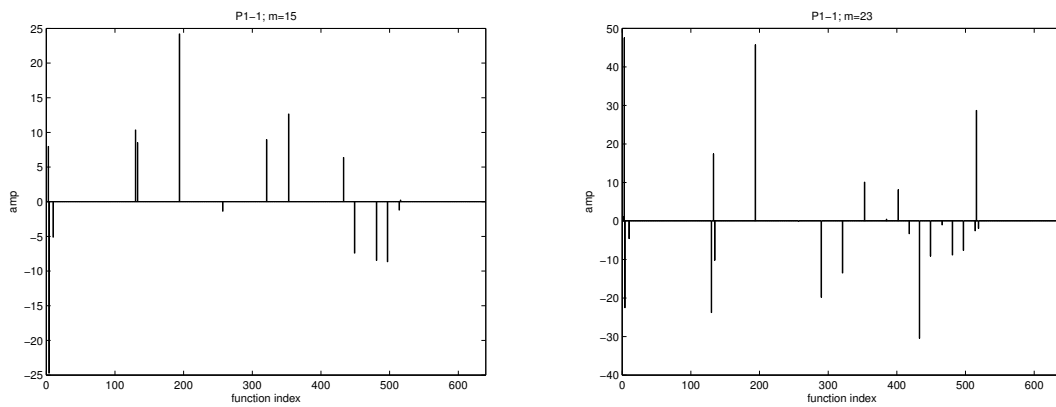


Figure 4.18: Signal decomposition solving (P_{1-1}) followed by a projection step. On the left: the amplitudes of the coefficients for the approximation with 15 elements. On the right: the amplitudes of the coefficients for the approximation with 23 elements.

signal and a particular dictionary. However they show how BPDN is not always superior to greedy algorithms and how thresholding can still have interesting performances. It is not easy to make comparisons concerning the computational complexity, since they strongly depend on the technique adopted to solve (P_{2-1}) and (P_{1-1}) . In general, while thresholding turns out to be the faster method, greedy algorithms require less computational load than LP and QP strategies.

Table 4.1: MSE obtained by all the presented methods for the approximation presented in Sec. 4.3.1.

| m | Threshold | MP | OMP | (P_{2-1}) | $(P_{2-1}) + \text{Proj}$ | (P_{1-1}) |
|-----|-----------|-------|-------|-------------|---------------------------|-------------|
| 15 | 0.080 | 0.016 | 0.012 | 0.448 | 0.078 | 0.170 |
| 23 | 0.071 | 0.003 | 0.003 | 0.107 | 0.009 | 0.077 |

4.7 Other Methods

In this chapter we have seen many approaches to the problem of signal decomposition. In particular we presented several techniques for selecting few functions among the ones that compose a redundant dictionary. The selected atoms should be able to give an approximation as sparse as possible. Of course there are other possible approaches and in this last section we briefly present three other interesting methods for signal decomposition. For these no numerical example will be provided.

4.7.1 Method of Frames

A very common technique for function selection is the Method of Frames (MOF) [39], or Least Squares. Among all the possible solutions of Equation (4.2), the MOF picks out the one that has a minimal ℓ_2 norm:

$$\mathbf{c}_{MOF} = \arg \min_{\mathbf{c} \in \mathbb{R}^d} \|\mathbf{c}\|_2 \quad \text{s.t.} \quad D\mathbf{c} = f. \quad (4.55)$$

The solution to this problem is unique and can be expressed as:

$$\mathbf{c}_{MOF} = D^+ f = D^T (DD^T)^{-1} f. \quad (4.56)$$

In general, the numerical computation of the pseudo-inverse in (4.56) involves expensive operations like the Singular Value Decomposition (SVD) and has a complexity of the order of $O(n^3)$. However, if D is not ill-conditioned, there exist faster techniques based on the conjugate gradient method. In the case of “tight frames” dictionaries, the solution of the MOF can be expressed in a closed form and can be calculated in about $n \log_2 n$ operations.

In an approximation framework we can write the MOF as:

$$(P_{2-2}) \quad \min_{\mathbf{c} \in \mathbb{R}^d} \|f - D\mathbf{c}\|_2^2 + \gamma \|\mathbf{c}\|_2^2. \quad (4.57)$$

This method was frequently used, since it has low computational cost, but generally results obtained by the MOF are far from being sparse. This can be simply explained since, as saw in Section 4.1.2, the ℓ_2 norm is not a sparsity-preserving function. Figure 4.7, in the graphic on top right, shows a clear example of a situation where the MOF fails in finding a sparse representation of a simple signal.

4.7.2 Best Orthogonal Basis

For specific dictionaries, it is possible to develop decomposition techniques custom-tailored to the dictionary. Two examples are wavelet packet and cosine packet, that can be seen as a union of many orthogonal bases. In [37] Coifman and Wickerhauser have presented a method of adaptively select a single basis from among all the many orthogonal bases that compose the dictionary. The selected basis is the “best” one, and this gives the name to the algorithm: Best Orthogonal Basis (BOB). There exists a fast algorithm to select the best basis, whose complexity is $O(n \log_2(n))$.

When the signal f has a sparse representation in one orthogonal basis of the dictionary BOB is in general able to give a near-optimal decomposition. If this is not the case the performances of BOB definitively decreases.

4.7.3 FOCUSS

Another interesting approach for solving exact representation problem is a variation of the MOF called FOCal Underdetermined System Solver (FOCUSS) [82]. This is a non-parametric, iterative algorithm for finding localized solutions to undetermined problems with limited data. It is composed of two main parts:

1. Retrieval of a low resolution estimate of the sparse signal by means of a simple MOF approach.
2. Pruning process of the first estimation using a generalized affine scaling transformation. That is, an iterated solution is found by scaling the entries with the solution of previous iterations.

Formally, at every iteration k the following problem is solved:

$$\arg \min_{\mathbf{b}} \|(W_{a_k} W_{p_k})^+ \mathbf{b}\|_2^2 \quad \text{s.t.} \quad D\mathbf{b} = f, \quad (4.58)$$

where W_{a_k} is a diagonal matrix containing some *a priori*, and W_{p_k} is another diagonal matrix composed the weights obtained from the solution retrieved in the precedent iteration. The procedure to solve (4.58) is:

$$\begin{aligned} \text{Step I: } & W_{p_k} = \text{diag}(\mathbf{b}_{k-1}^l) \\ \text{Step II: } & q_k = (DW_{a_k} W_{p_k})^+ f \\ \text{Step III: } & b_k = W_{a_k} W_{p_k} q_k, \end{aligned} \quad (4.59)$$

where W_{p_0} is assumed to be the identity matrix and l is an user defined parameter which modifies the strength of the re-weighting feed-back. Note that for $l = 1/2$, FOCUSS provides the solution to the BP problem (see [42, 43]).

If one sees the weights in W_{p_k} as some kind of *a priori* knowledge about the solution, then an interpretation can be that the algorithm computes its own *a priori* information from iteration to iteration.

The version of the algorithm we just described is suited for exact representation. However, there is a slight modification of FOCUSS which allows to find signal approximations. This can be found in [81, 82].

Image and Video Coding using Redundant Dictionaries

Coding an image is difficult because images are complex and catching their structure is a hard task which entails an elaborate analysis. This was pointed out in the previous chapters. What we can add here is the fact that the use of *ad-hoc* redundant dictionaries, if from one side allows a much more flexible representation, on the other side makes the coding procedure even more complex. Moreover, the relation between sparse approximation and efficient compression is not straightforward. In fact we do not only need (quantized) coefficients, but it is also necessary to specify which functions these coefficients refer to. This task in general requires more information as the size of the dictionary increases. Thus the importance of combining approximation techniques with appropriate coding strategies in order to obtain an efficient compression is evident [36].

This chapter faces the problem of low bit-rate coding for both still pictures and videos. This is done by using redundant dictionaries and the function selection tools illustrated in the first part of the dissertation.

Why do we care about lossy compression? This is a natural question, since storage capacities and networks speeds are growing more and more quickly. Just think that few months ago a mobile phone achieved 1 giga bit-per-second real-time packet transmission in down-link [4]. And this moving at about 20 km/h!

The first point motivating lossy image and video compression is our imperfect vision system, which is not able to detect all the information contained in a picture (still or moving). Nevertheless, this does not extend to very low bit-rate coding, where the information loss is undoubtedly perceptible. We investigated in this direction for three main reasons. Firstly, it is a more challenging problem since the high bit-rate compression is nowadays almost a dismissed case for the latest codecs. Secondly, it is matching the signal analysis strategies we adopted carrying out this research and it is close to the principle of sparseness. Finally, low bit-rate compression is still relevant for certain classes of applications. Among them we can cite video surveillance or compression of identity pictures for very low memory devices. In general, additionally, if compression alone is losing interest, its association with feature extraction is lively and more and more important. Consider for example a large database where images are at the same time compressed and easy to detect thanks to their representation.

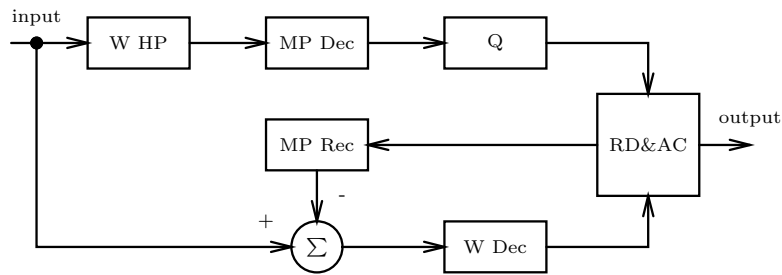


Figure 5.1: Encoding scheme: W HP is the high-pass filter using Wavelets, MP Dec and MP Rec are respectively the Matching Pursuit decomposition and reconstruction, Q represents the quantization operation, W Dec is the Wavelet decomposition, RD is the rate-distortion optimization and AC stands for Arithmetic Coding.

5.1 Still Image Compression

The state of the art in image compression is based on transform coding using orthonormal basis such as DCT or wavelets. These schemes have achieved high compression ratios thanks to the huge research work that has been performed in efficiently coding the transform coefficients and parameters. Nevertheless these traditional approaches suffer some severe limitations. As shown in Chapter 2, bidimensional wavelets, for example, fail to capture regularities of contours, since they are not able to sparsely represent one-dimensional singularities of 2-D signals. We have already seen that many research efforts were aimed at representing a natural image exploiting its inherent geometrical structure. Only very few of them, up to now, have given results in terms of compression. In particular, the most relevant approaches that follow this idea can be found in the already cited works [181], [162] and [110].

In this section we aim at obtaining an efficient encoding of natural images by approximating the contours by a sum of bidimensional, non-separable functions. The residual, that we suppose constituted by the smooth part of the image and textures, is then coded using wavelets. This representation method, illustrated in Figure 5.1, follows the image model introduced in Chapter 3 that we repeat here for the ease of the reader:

$$I \simeq I_{smooth} + I_{edge} + I_{texture}. \quad (5.1)$$

The dictionary, described in Section 5.1.1, has been designed to match the object contours. These discontinuities have most of their energy at high frequencies. Therefore, before coding the edges, the image is decomposed with wavelets and reconstructed keeping all the subbands but the low-pass. This step is equivalent to a high-pass filtering and it is labeled as “W HP” in Fig. 5.1. The high frequency content of the signal is thus decomposed over the edge-oriented dictionary using the Matching Pursuit algorithm. After that MP projection coefficients have been quantized, a residual image is computed by subtracting the quantized MP reconstruction from the original input image. This residual contains the low frequencies of the signal, the textures and the artifacts introduced by MP (the latter also include quantization errors). As can be seen in the scheme of Fig. 5.1 the residual is decomposed with wavelets. The wavelet functions, used for both decomposing the residual and computing the high-frequency input for MP, are the Cohen-Daubechies-Feauveau 9,7 [35]. At this point we have low-pass wavelet coefficients (projections on the scaling function) representing I_{smooth} , atoms from the edge-oriented dictionary representing I_{edge} and the high frequency wavelet coefficients representing $I_{texture}$ and correcting the MP artifacts, if any. Note that the quantization

of the MP indexes and projections is fixed. The coefficients of the coarse version representing the smooth part of the image are quantized in a differential way (DPCM), while the wavelet coefficients are subject to a deadzone quantization with the deadzone step twice the quantization step. The quantization steps for the coarse signal and wavelet coefficients are independent.

All these parameters are subject to a rate-distortion optimization that establishes the number of atoms to code, the quantization step for the DPCM of the coarse version and the step for the wavelet deadzone quantization. All the parameters and the quantized coefficients are entropy coded using an adaptive arithmetic coding algorithm [79].

5.1.1 The Dictionary

We use a multi-component dictionary resulting of the union of the Cohen-Daubechies-Feauveau 9,7 biorthogonal wavelet basis with an edge-oriented dictionary. The latter is built by anisotropically scaling, orienting and bending a generating function, resulting in an overcomplete basis set. Such dictionary was illustrated in detail in Section 3.2. The two generating functions are described in Equation (3.46). Figure 3.9 shows five atoms of the edge-oriented dictionary in both space and frequency domain, illustrating also the effect of bending, rotation and anisotropic scaling.

Finally we obtain a highly redundant dictionary, with a redundancy factor $s \simeq 17000$. The size of the dictionary increases dramatically allowing big scaling factor along x_2 , indeed the number of rotations is proportional to the scale parameter a_2 . A dictionary including elongated atoms is able to better represents long edge structures.

Even if big dictionaries can be built with a small coherence [92], our dictionary has high coherence since we adopt a geometric oriented design. Thus, we cannot theoretically assure that MP recovers the best sparse approximation of the signal. Nevertheless we notice a fast energy decay of the residual at first iterations, which means that the dictionary copes well with natural data. MP is able to select good atoms, at least during first iterations.

5.1.2 Searching Algorithm

A greedy algorithm is used to decompose the detail version of the image in its most important features. As seen in Section 4.4, this selects at each iteration an atom from the dictionary such that the projection coefficient $|\langle g_{i_k}, r_k \rangle|$ is maximum. To find such g_{i_k} we use a full search algorithm that computes the inner products between the residual and all the functions of the dictionary. Since the dictionary is composed of all the translations of the transformed generating functions (TGF), it is clear that all the inner products between the TGF translated all over the residual and the residual itself, correspond to the convolution of the TGF with the residual. To speed up the search, we compute the convolutions as products in the frequency domain. To avoid problems with the regions at the border of the image a padding of ten pixels is added. The Fourier transform of all the TGF is computed only once and stored.

The complexity of a MP decomposition of a signal of n samples results to be of the order of

$$C \cdot m \cdot s \cdot n \log_2 n, \quad (5.2)$$

where m is the number of chosen atoms, the constant C depends on the strategy adopted for atom selection and s depends on the size of the dictionary. In fact, $s = |\mathcal{D}|/n$ is the redundancy of the dictionary and it corresponds to the size of \mathcal{D} without considering translations. Note that in our implementation s does not depend on n ! In particular we use the modified version of Matching Pursuit illustrated in Section 4.4.3, which at each iteration selects more than one quasi-orthogonal

atoms. This algorithm turns out to be much less computationally expensive, having in Eq. (5.2) a constant $C \ll 1$.

In order to further speed up the atom selection, another algorithm, based on a tree-based pursuit decomposition, may be taken into account [104]. But since the quality loss is not negligible, especially in the range of bit-rate we are interested in, we decided not to use this searching method.

5.1.3 Rate-Distortion Optimization

The choice of the number of atoms to code and the quantization step for the wavelet coefficients is based on a rate-distortion optimization. In order to study the RD of our representation method, we take into account the image model expressed in Equation (5.1).

We indicate with $\tilde{I}_{edge}(m)$ the m -term approximation of the edge part of the image obtained by the MP decomposition of I_{edge} over the contour oriented dictionary. Thereby, the input of the wavelet decomposition (I_{st}), supposed to contain the smooth and texture part, is obtained by subtracting \tilde{I}_{edge} from the original image:

$$I_{st}(m) = I - \tilde{I}_{edge}(m) = I_{smooth} + I_{texture} + (I_{edge} - \tilde{I}_{edge}(m)). \quad (5.3)$$

Since $\tilde{I}_{edge}(m)$ is the superposition of m atoms with quantized projections, the final error is given by

$$I_{err} = I - \tilde{I}_{st}(m, \Delta), \quad (5.4)$$

where $\tilde{I}_{st}(m, \Delta)$ is the approximation of $I_{st}(m)$ given by the quantization of the wavelet coefficients. Let R_{MP} the rate due to the atoms and R_W the one due to the wavelet coefficients, the total rate is $R = R_{MP} + R_W$. It depends on the number of atoms used to approximate I_{edge} and on the quantization steps of the wavelet coefficients. Before investigating the rate-distortion of our representation method, we study the rate related to the MP expansion, and we recall the RD theory concerning wavelet coding.

MP Rate

Our signal approximation over \mathcal{D} is represented by the atom indexes, positions and projections. The indexes or parameters that characterize the atoms shape are entropy coded using an adaptive arithmetic coding algorithm. We choose to make the x_2 -scale parameter depend on the radius used for bending the atoms. Therefore, the arithmetic coder uses the conditional probability $p(x_2\text{-scale}|\text{radius})$ to code the x_2 -scale, and $p(\text{rotation}|x_2\text{-scale})$ for the rotation parameter. In order to code the positions and projection coefficients, two different approaches can be taken into account. The first one consists in ordering the atoms in decreasing absolute projection values; then the projections can be quantized either by using an exponential quantizer [75] or in a differential way (DPCM) as done in [146]. The quantization is followed by arithmetic coding. The x_1 and x_2 coordinates of the atoms positions are then simply stored without any particular coding scheme.

The second approach performs a different sorting of the atoms in such a way to take advantage in coding their positions [133]. The atoms are ordered by raster scanning, then the x_1 and x_2 coordinates are coded in a differential way followed by arithmetic coding. The drawback is that the SNR scalability is lost [69, 75]. In this case a simple uniform quantization and arithmetic coding of the projections is performed.

As shown on the left-hand side of Figure 5.2, the position oriented coding method outperforms the projection coding, thus we chose to use the former to code atoms positions and projections. Finally the right-hand side of Figure 5.2 gives the total bit-rate per atom $R_a(m)$ as a function of the number of atoms used to approximate the edge component I_{edge} . The exponential decay of

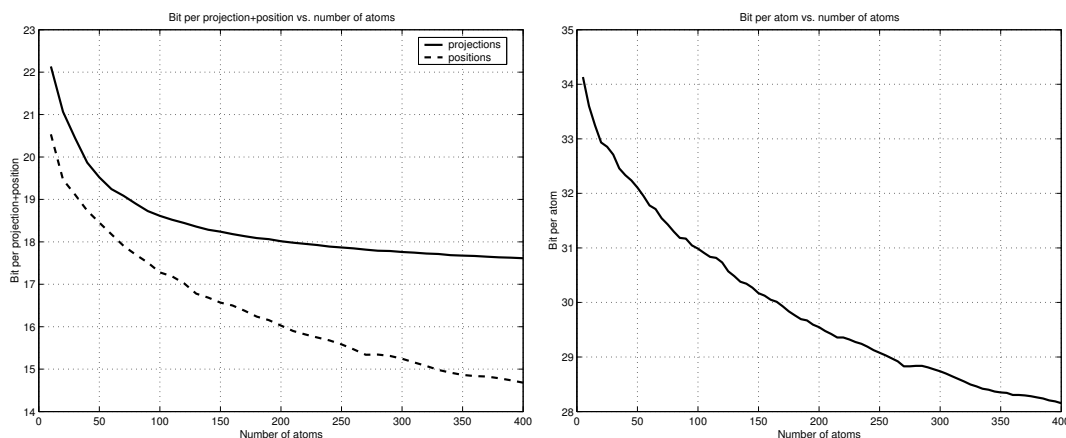


Figure 5.2: Left: Comparison between projections oriented and positions oriented coding. Right: Total number of bits per atom as a function of the number of atoms used to approximate I_{edge} . The position oriented method is used to code positions and projections.

$R_a(m)$ is due to the fact that increasing m , the grid representing the atoms positions becomes more dense and the entropy of the displacements between adjacent atoms decreases. At the limit we can say that the bits required to code an atom get close to the coding rate of the projection and shape parameters.

Wavelet Rate-Distortion

The wavelet functions used to decompose the residual are the Cohen-Daubechies-Feauveau 9,7 with normalization $(\sqrt{2}, \sqrt{2})$. This biorthogonal wavelet basis is nearly orthogonal and thus we suppose that the distortion given by the quantization in the wavelet domain coincides with the distortion in the original domain. In order to have a hint on the real rate-distortion behavior, let us make the hypothesis of high resolution quantization, although it is not always satisfied in the compression domain. Using a uniform quantizer, we can approximate the distortion or MSE as a function of the quantization step Δ (see [118]):

$$D_W = \frac{\Delta^2}{12}. \quad (5.5)$$

The rate, that corresponds to the entropy of the output indexes, depends on the quantization step and on the differential entropy

$$R_W = \frac{\sum_k N_k (h_k - \log_2 \Delta)}{n} = \frac{\sum_k N_k h_k}{n} - \log_2 \Delta, \quad (5.6)$$

where h_k is the differential entropy, N_k is the number of wavelet coefficients at resolution k and n is the size of the signal.

MP+Wavelet Rate-Distortion

Now we can formulate the rate-distortion of our coder based on MP and wavelet decomposition. The final distortion depends on the quantization step of the wavelet coefficients, and for fine quantization

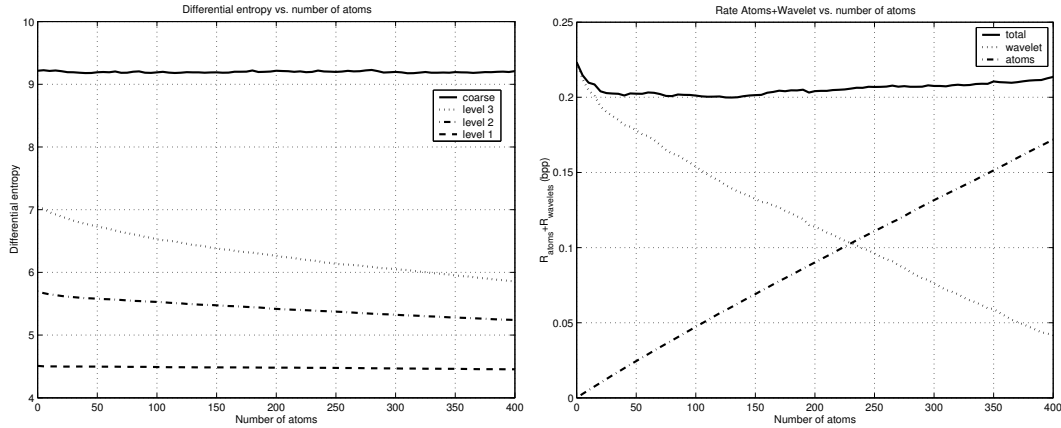


Figure 5.3: Left: Differential entropy of the wavelet coefficients at different resolution levels. Right: bpp of our representation method as a functions of the number of atoms. The quantization step is fixed, and so the final distortion.

we have that $D = D_W = \frac{\Delta^2}{12}$. Selecting m atoms, the total rate $R = R_{MP}(m) + R_W(m, \Delta)$ depends on the number of atoms approximating I_{edge} and obviously on the quantization step Δ ,

$$R_{MP}(m) = \frac{R_a(m)m}{n}, \quad (5.7)$$

$$R_W(m, \Delta) = \frac{\sum N_k h_k(m)}{n} - \log_2 \Delta. \quad (5.8)$$

It is important to notice that the differential entropy $h_k(m)$ associated to wavelet resolution level k , depends on the number of atoms that represent $\tilde{I}_{edge}(m)$. Indeed the statistic of $I_{st}(m) = I - \tilde{I}_{edge}(m)$ changes with m , especially at the resolution levels that contain the energy of the edge structures. The left-hand side of Figure 5.3 illustrates how the differential entropy depends on m . When an atom is subtracted from the image, the entropy decays in all the subbands except level 1 (the high frequencies) where it stays more or less constant. The reason is because at high frequencies the influence of the atoms is very small: let us say, even if it is not completely exact, that this is the range of textures. One can also observe how, in general, the impact of subtracting an atom decreases with iterations, which intuitively corresponds to the exponential decay of the residual energy discussed in [120].

We now write the Lagrangian cost as

$$L(m, \Delta) = D(\Delta) + \lambda (R(m, \Delta) - R_{bpp}) = \frac{\Delta^2}{12} + \lambda \left(\frac{R_a(m)m}{n} + \sum_k \frac{N_k}{n} h_w(m) - \log_2 \Delta - R_{bpp} \right), \quad (5.9)$$

where R_{bpp} is the bit budget per pixel. Differentiating with respect to Δ and m (neglecting the integer constraint on m), we obtain

$$\frac{\partial L}{\partial \Delta} = \frac{2\Delta}{12} - \frac{\lambda}{\Delta \ln 2}, \quad (5.10)$$

$$\frac{\partial L}{\partial m} = \lambda \left(\frac{R'_a(m)m + R_a(m)}{n} + \sum_k \frac{N_k}{n} h'_w(m) \right) \quad (5.11)$$

and setting the derivative to 0 we have

$$\Delta = \sqrt{\frac{6\lambda}{\ln 2}} \quad (5.12)$$

and

$$R'_a(m)m + R_a(m) + \sum_k N_k h'_k(m) = 0. \quad (5.13)$$

Solving Equation (5.13) we find the optimal number of atoms that minimizes the rate for a given distortion. It is important to notice that the solution of (5.13) does not depend on the final distortion D . This is due to the assumption of fine quantization of the wavelet coefficients. Once we get the optimal number of atoms m_{opt} , setting to zero the derivative of Equation (5.9) with respect to λ , we obtain the quantization step as a function of the bit budget,

$$\Delta = 2^{\left(\frac{R_a(m_{opt})m_{opt}}{n} + \sum \frac{N_k}{n} h_k(m_{opt}) - R_{bpp}\right)}. \quad (5.14)$$

Remark that this is true only at high bit-rate, and that the functions $R_a(m)$ and $h_k(m)$ have to be estimated. Figure 5.3 shows on the left-hand side the behavior of the differential entropy at different resolution levels, and on the right-hand side shows the total rate as a function of m for a fixed step Δ (changing Δ corresponds to a vertical translation of the wavelet rate). The minimum rate is reached coding m_{opt} atoms: for the studied examples this minimum occurs between 100 and 150 atoms.

In practice, at low bit-rates the fine quantization hypothesis is not satisfied and the simple model for the wavelet RD does not fit its real behavior. Moreover, we use a deadzone uniform quantizer, which improves the rate distortion at low bit-rate quantization [119], and different steps of quantization for the coarse and wavelet coefficients can be chosen. Implementing a numerical rate distortion optimization, it turns out that the optimal number of atoms changes depending on the bit-rate. All the graphics in this section show the mean of the results obtained using three standard images *Lena*, *cameraman* and *peppers*.

5.1.4 Results and Comparisons

Combining the MP approximation properties with an accurate design of the dictionary makes it possible to achieve high compression ratios, catching the most visually relevant structures of natural images.

This section provides some results obtained with the presented algorithm; in the following, for the sake of simplicity, it will be shortly called MPW. A comparison is made with the standard JPEG2000, following the implementation in [8]. Another point of comparison is a pure wavelet encoder we have developed which uses exactly the same coding options and RD optimization we adopted in the MPW coding scheme. In the following, we refer to it as “Wavelets”. Table 5.1 shows the PSNR vs. bit-rate results for the images *peppers* and *cameraman*. Both images have size 256×256 pixels. For this size, the computational time for MPW coding is around one hour, using a 2GHz processor.

At very low bit-rates, our algorithm obtains good results because it is capable to catch the main features of a natural image with few functions. It is fair to observe that at less than 0.1 bpp the gap between JPEG2000 and MPW is also partly due to the bigger size of the JPEG2000 header: in fact one can notice that even our very simple “Wavelets” encoder outperforms the standard. The size of the JPG2000 header is 148 byte, meanwhile MPW has a simple header of size 22 byte. Regarding JPEG2000, if we take into account only the bit stream due to the quantized wavelets coefficients, we observe an increase of about 0.6 dB, 0.4 dB and 0.2 dB respectively at 0.1 bpp, 0.2 bpp and 0.3

Table 5.1: PSNR vs. bit-rate for the images *cameraman* and *peppers*: comparisons between MPW, JPEG2000 and a coding scheme based on DWT.

| cameraman (256×256) | | | | peppers (256×256) | | | |
|--------------------------------|-------|----------|----------|------------------------------|-------|----------|----------|
| Rate (bpp) | MPW | JPEG2000 | Wavelets | Rate (bpp) | MPW | JPEG2000 | Wavelets |
| 0.05 | 22.50 | 21.00 | 21.50 | 0.06 | 23.13 | 21.36 | 22.10 |
| 0.10 | 25.06 | 23.63 | 23.79 | 0.10 | 25.33 | 23.83 | 23.90 |
| 0.15 | 26.45 | 25.23 | 24.96 | 0.15 | 27.03 | 25.89 | 25.68 |
| 0.20 | 27.38 | 26.45 | 26.10 | 0.20 | 28.25 | 27.25 | 27.00 |
| 0.25 | 28.11 | 27.38 | 27.02 | 0.25 | 29.09 | 28.62 | 28.05 |
| 0.30 | 28.76 | 28.53 | 27.80 | 0.30 | 29.91 | 29.79 | 29.00 |
| 0.35 | 29.27 | 29.57 | 28.51 | 0.35 | 30.56 | 30.62 | 29.80 |
| 0.50 | 30.61 | 31.16 | 30.21 | 0.50 | 32.24 | 32.85 | 31.89 |

bpp. This difference between JPEG2000 and MPW disappears for bigger images (e.g. 512×512 pixels). In general, observing Figures 5.4, 5.5, 5.6 and 5.7, one can see that the proposed scheme outperforms JPEG2000 not only in terms of PSNR but also and especially of visual quality. But some problems also arises. Figure 5.4 (bottom left) shows some artifacts presented by the proposed MPW method: at very low bit-rate edges can be not completely coded. Other kinds of artifacts can be seen in Figure 5.6 (bottom left): again concerning contours, it can happen that edge-oriented atoms are misplaced. In addition the same picture presents wavelets artifacts along contours (e.g. look at the arm) that came from the use of the “W HP” filter at the beginning of the coding scheme. Such problem can be strongly reduced by using better high-pass filters, as shown in the next section. Another visual annoying element in the same picture is given by the background, but it can be solved by using a finer quantization for the low-pass coefficients.

We compare results up to 0.5 bpp, where there is no relevant visual difference between the images compressed with our method and JPEG2000. In addition, at higher bit-rates, the original and compressed images are visually identical. Results for a bigger image are shown in Figure 5.7, where a picture of *Lena* [163] of size 512×512 is compressed at 0.1 bpp. All the test images are available in my web page (<http://lts2www.epfl.ch/~granai/research.htm>) where one can also find further compression results and comparisons.



Figure 5.4: Comparison between JPEG2000 and MPW at about 0.05 bpp. Top: original peppers image (256×256). Bottom left: image compressed with MPW (PSNR = 23.13). Bottom right: image compressed with JPEG2000 (PSNR = 21.36).

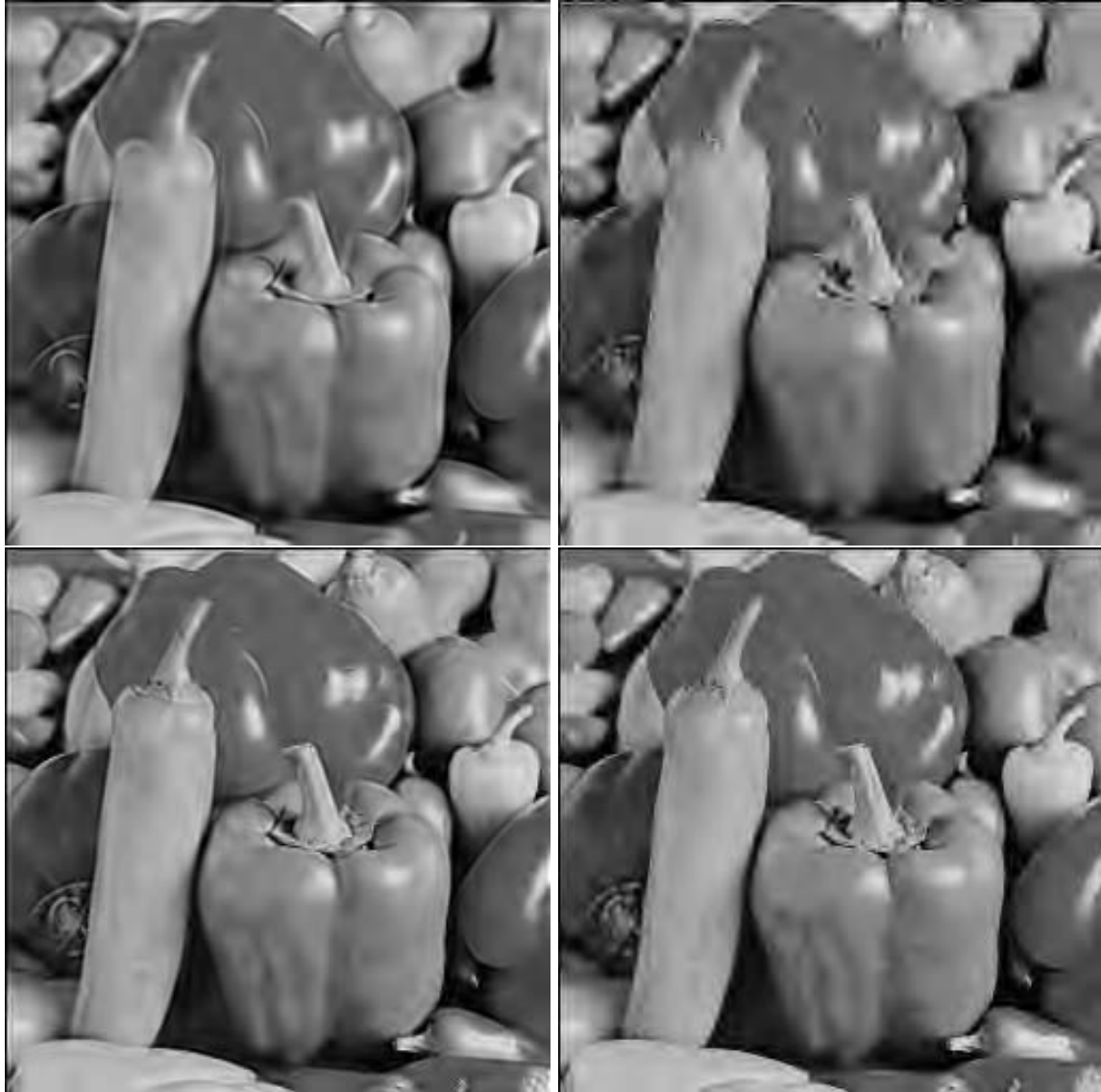


Figure 5.5: Comparison between JPEG2000 and MPW at 0.15 (top) and 0.3 (bottom) bpp. Top left: image compressed with MPW at 0.15 bpp (PSNR = 27.03). Top right: image compressed with JPEG2000 at 0.15 bpp (PSNR = 25.89). Bottom left: image compressed with MPW at 0.3 bpp (PSNR = 29.91). Bottom right: image compressed with JPEG2000 at 0.3 bpp (PSNR = 29.79) The original image can be seen on the top of Fig. 5.4



Figure 5.6: Comparison between JPEG2000 and MPW at 0.15 bpp. Top: original cameraman image (256×256). Bottom left: image compressed with MPW. Bottom right: image compressed with JPEG2000.



Figure 5.7: Comparison between JPEG2000 and MPW at 0.1 bpp for the image Lena of size 512×512 (on the top). Bottom left: image compressed with MPW (PSNR = 29.84). Bottom right: image compressed with JPEG2000 (PSNR = 30.00).

5.1.5 Reprise

In [146] we proposed a similar but less elaborate coding approach also based on Matching Pursuit. The main difference is that this time textures are not taken into account and the approximation wants to capture only smooth parts and edges. This can be effective at and only at very low bit-rates. The coding scheme, already introduced in Section 3.3 simply consists on deriving from an original image a coarse approximation by low-pass filtering and downsampling. Based on this coarse version, it predicts the original by upsampling and filtering and calculates the difference as the high-pass version or detail version. The detail version is then represented using the atoms from the edge-oriented dictionary on the model of the one illustrated in Section 3.2. Figure 5.8 illustrates this alternative compression scheme. Details on the filters $A(\vec{\omega})$ and $B(\vec{\omega})$ can be found in Sec. 3.3 or directly in [146].

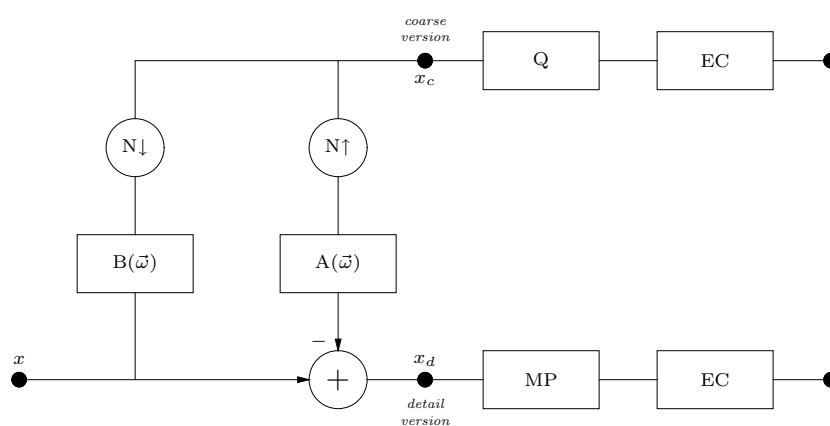


Figure 5.8: Laplacian Pyramid. Only the encoding part is shown. $B(\vec{\omega})$ is the 2-D low-pass filter, $A(\vec{\omega})$ is the 2-D interpolation filter, N is the downsampling factor. Q represents the quantization operation, MP the Matching Pursuit and EC stands for Entropy Coding.

Other minor differences between these two approaches are that the edge-oriented overcomplete dictionary in MPW has a slightly bigger redundancy factor and the generating functions are not exactly the same (see again Section 3.2). Moreover the projection coefficients of the selected atoms are ordered in a decreasing order and quantized in a differential way (DPCM).

The results obtained at very low bit-rates with this second approach are quite similar but slightly worse in terms of PSNR as can be seen in Table 5.2, where they are compared with JPEG2000. Figures 5.9 and 5.10 show some compressed images that visually illustrates the advantage of the proposed method at very low bit-rates. Thanks to the use of the wavelets for coding the residual (see Figure 5.1), the gain of MPW is not only limited at very low bit-rates.

Table 5.2: PSNR vs. bit-rate for the images *cameraman* and *Lena*: comparison between the proposed algorithm based on MP and JPEG2000.

| cameraman (256 × 256) | | | Lena (256 × 256) | | |
|-----------------------|-------|----------|------------------|-------|----------|
| Rate (bpp) | MP | JPEG2000 | Rate (bpp) | MP | JPEG2000 |
| 0.030 | 20.45 | 18.41 | 0.036 | 22.43 | 21.02 |
| 0.052 | 22.46 | 21.07 | 0.062 | 24.71 | 23.40 |
| 0.077 | 23.92 | 22.62 | 0.079 | 25.51 | 24.32 |
| 0.100 | 24.75 | 23.63 | 0.100 | 26.30 | 25.28 |
| 0.125 | 25.46 | 24.53 | 0.125 | 27.15 | 26.14 |
| 0.147 | 25.61 | 25.13 | 0.150 | 27.70 | 26.89 |



Figure 5.9: Comparison between JPEG2000 and the proposed scheme based on MP at 0.079 bpp for the image *Lena* of size 256×256 . Left: image compressed with MP (PSNR = 25.51). Right: image compressed with JPEG2000 (PSNR = 24.32).



Figure 5.10: Comparison between JPEG2000 and the proposed scheme based on MP at 0.077 (top) and 0.147 (bottom) bpp. Top left: image compressed with MP at 0.077 bpp (PSNR = 23.92). Top right: image compressed with JPEG2000 at 0.077 bpp (PSNR = 22.62). Bottom left: image compressed with MP at 0.147 bpp (PSNR = 25.61). Bottom right: image compressed with JPEG2000 at 0.3 bpp (PSNR = 25.13). The original image can be seen on the top of Fig. 5.6

5.2 Video Coding

Motion video data is essentially a time-ordered sequence of pictures. The most successful class of video compression algorithms is based on hybrid methods consisting in the combination of prediction loops in the temporal dimension (motion estimation/motion compensation) with a suitable uncorrelation technique in the spatial domain (transform coder), as illustrated in Figure 5.11. For coding purposes, quantization and dequantization are inserted after the 2-D transform and before the inverse 2-D transform respectively. Highly nonlinear predictors are used in order to adapt the representations as much as possible to the structure of video signals. As a major tool, motion compensation is used to capture and represent efficiently temporal video geometric changes. Often in video signals, few motion parameters are able to model frame to frame changes (up to some accuracy) and, thus, supply good frame approximations that generate small residual errors when used within hybrid predictive video representations. Commonly, simple translational models together with block matching are used in predictive video coding (but many efforts have been spent also in other directions).

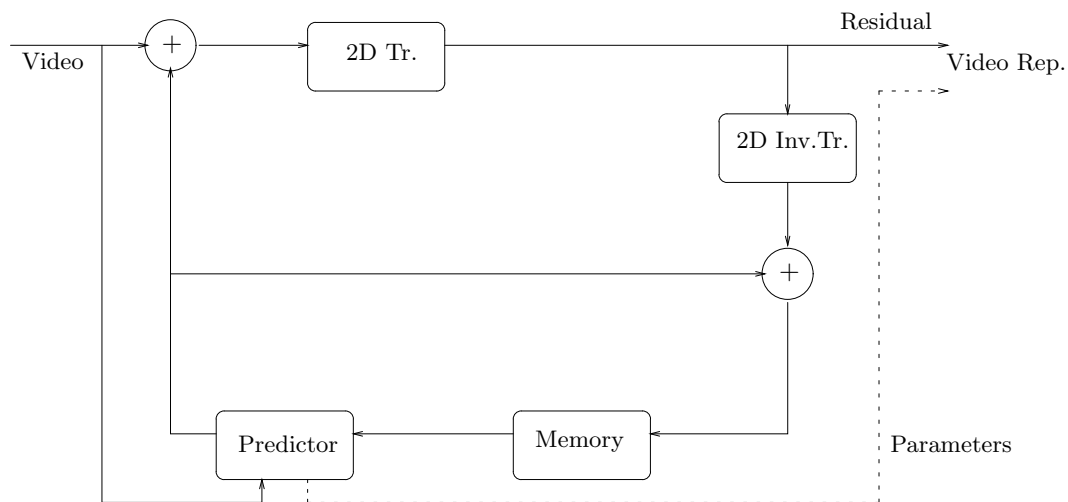


Figure 5.11: Block diagram of a predictive video representation scheme.

The state of the art for hybrid video coding is specified by the recent standard H.264, also named Advanced Video Coding (AVC) (ITU-T Rec. H.264, or ISO MPEG-4, part 10).

In this section we present a work aimed at exploiting the advantages of coding the Displaced Frame Difference (DFD), output of the motion compensation (MC) algorithm, using a redundant dictionary. In order to remain as close as possible to the state of the art, we adopt a motion estimation that is compatible with H.264 (see Section 5.2.1). The output of this block is then coded using a pursuit algorithm and an appositely designed bidimensional, anisotropic dictionary. Thanks to this technique we achieve a sparse representation of the signal and therefore a more compact energy concentration.

5.2.1 Motion Estimation

High compression efficiency in video coding is achieved by adopting hybrid systems which combine two stages. In the first stage motion estimation and compensation predict each frame from the neighboring frames. At the second one the prediction error is coded. Current video compression

standards use block-based orthogonal transforms to code the residual error. These two stages are then followed by appropriate entropy coding.

Relative to prior coding methods, the standard H.264/AVC has an enhanced motion estimation that allows higher compression ratios [183]. In particular we can attribute this improvement to the new variable block-size motion compensation with small block sizes, the quarter-sample-accurate motion compensation and the use of multiple reference frames. Moreover the 4x4 integer transform turns out to be well adapted to this kind of motion compensation [121].

In the proposed coding scheme, we adopt some of the new features introduced by this standard and obtain a motion compensation scheme that is compatible with H.264. In particular the following features are used:

- variable block-size motion compensation, with a minimum size of 4x4,
- tree-based MC,
- MC with quarter-pel accuracy,
- use of improved “skipped” motion inference [183].

Our encoder allows I and P-pictures only. An I picture (or I-frame, short for intraframe, also called keyframe) is a single frame of a video that the compressor examines independent of the frames that precede and follow it and stores all of the data needed to display that frame. This means that there is no reduction of the temporal redundancy. I-frames allows to stop the error propagation due to the motion estimation and can have a better quality but the price is a much higher amount of bit required for their coding. A P-frame is encoded relative to the past reference frame. A reference frame is a P or I-frame. For more details we refer to [173].

Moreover, due to the frame-based structure of our MP codec, intra-blocks are not permitted. I-pictures are fully compliant with the H.264/AVC standard, using the integer transform illustrated in [121]. Only three of the nine prediction directions are used and only the 4×4 predicted block mode is implemented (not the 16×16 one) [183].

5.2.2 Coding of Displaced Frame Differences

The residual error of the motion compensated prediction still contains spatial redundancy: to reduce the amount of resources needed for transmission, this error is typically coded via block-based DCT. In H.264/AVC, this transform is replaced by an integer orthogonal approximation of the DCT, able to work with 4x4 blocks and so compatible with the finest motion compensation segmentation. The advantage of this transform is that it can be computed exactly in integer arithmetic, so avoiding inverse transform mismatch problems; moreover, it reduces the computational complexity thanks to the fact that it can be calculated without multiplications, in 16-bit arithmetic [121].

However, linear invariant block-based transforms are far from optimal for representing (and then compressing) bidimensional signals such as natural images or motion compensated images [118]. In an important series of papers [15, 134, 135, 136] the authors have shown that improved coding efficiency can be achieved by replacing the DCT with an overcomplete non-orthogonal transform. This kind of approach, together with a suitable dictionary design, can represent a valid alternative to DCT or wavelet based schemes, especially (but not necessarily only) at low bit-rates, where most of the signal energy can be captured by only a few elements of the dictionary.

In the proposed scheme, the output of the motion estimation is a predicted image that is subtracted from the current frame. The DFD, difference between these two images, is then coded by using atoms selected over a redundant dictionary. Such selection is performed using the Matching

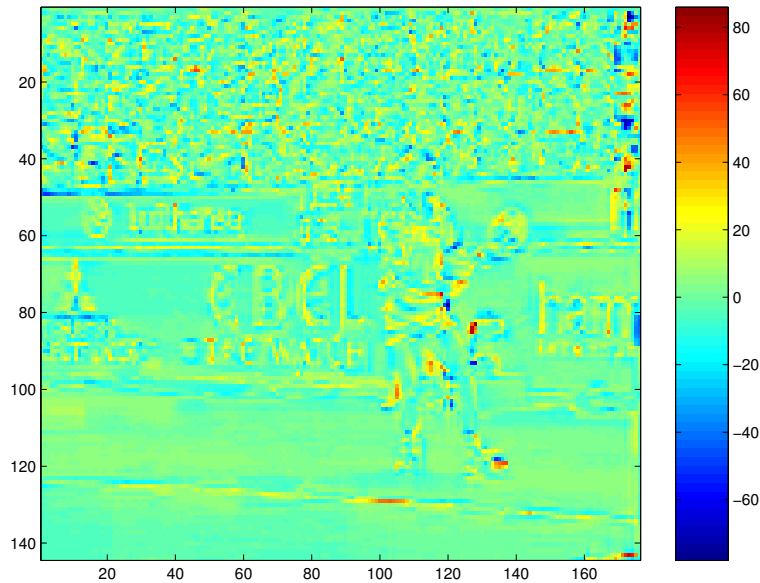


Figure 5.12: A displaced frame difference output of the motion compensation of H.256/AVC. From the sequence *Stefan* in QCIF format.

Pursuit algorithm already described in Section 4.4. Note that this method is not block-based: both the coding and the atom selection procedures work on the full frame, without any spatial subdivision.

5.2.3 Dictionary Design

Dictionary design is a crucial item for MP, since it strongly affects its convergence and visual performances. The dictionary used in our experiments is particularly suited for exploiting the signal structures of DFDs, mainly thanks to the use of peculiar generating functions and anisotropy (see also [130]). Figure 5.12 shows a DFD output of the motion compensation of H.264/AVC. The possible range of the pixel values is $[-255, 255]$, even if usually it turns out to be much smaller. Moreover a DFD commonly has a mean close to zero. Plainly, the structures of such kind of signals are very different from the ones analyzed in natural images: more specifically DFD pictures are, of course, much less correlated, more spiky and they present some artifacts due to the MC.

The dictionary (\mathcal{D}) used to decomposed the DFD is designed along the lines of the one for natural images, described in Section 3.2. It is thus composed of a set of real bidimensional functions, built by applying the following three types of transformations to the generating function $\phi(\vec{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}$ with $\vec{x} = (x_1, x_2)$.

- a) Translation $\mathcal{T}_{\vec{b}}$, to move the atom all over the frame:

$$\mathcal{T}_{\vec{b}} \phi(\vec{x}) = \phi(\vec{x} - \vec{b}). \quad (5.15)$$

- b) Rotation \mathcal{R}_{θ} , to locally orient the atom:

$$\mathcal{R}_{\theta} \phi(\vec{x}) = \phi(r_{\theta}(\vec{x})), \quad (5.16)$$

where r_{θ} is a rotation matrix

$$r_{\theta}(\vec{x}) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \quad (5.17)$$

c) Anisotropic scaling \mathcal{S}_{a_1, a_2} :

$$\mathcal{S}_{\vec{a}} \phi(\vec{x}) = \mathcal{S}_{a_1, a_2} \phi(x_1, x_2) = \phi\left(\frac{x_1}{a_1}, \frac{x_2}{a_2}\right). \quad (5.18)$$

Atoms are generated varying the parameters \vec{b}, θ, \vec{a} of the three previous transforms in the following order:

$$g_{(\vec{b}, \theta, \vec{a})}(\vec{x}) = \mathcal{T}_{\vec{b}} \mathcal{R}_{\theta} \mathcal{S}_{\vec{a}} \phi(\vec{x}). \quad (5.19)$$

Finally the obtained waveforms are normalized as follows:

$$g_{(\vec{b}, \theta, \vec{a})}^{\text{norm}}(\vec{x}) = \frac{g_{(\vec{b}, \theta, \vec{a})}(\vec{x})}{\|g_{(\vec{b}, \theta, \vec{a})}(\vec{x})\|_2}. \quad (5.20)$$

The dictionary used by the MP algorithm is obtained by uniformly discretizing the parameters θ and \vec{a} :

$$\mathcal{D} = \{g_{(\vec{b}, \theta, \vec{a})}^{\text{norm}}(\vec{x})\}_{\vec{b}, \theta, \vec{a}}. \quad (5.21)$$

We saw in the study of the dictionary for still images that bended atoms can improve the quality of the approximation. This option has been tested also for video signals, finding that only an extremely small gain in terms of error and visual quality is obtained, but with the drawback of a tremendous increase of the dictionary size. Thus, we chose not to include this transformation in our set.

The functions which generate the whole dictionary with the previous transformations have been selected in order to best match the characteristics of the input signal, i.e. the DFD coming out from the motion compensation block. In particular three functions have been chosen:

- A second derivative of a B-Spline on the x_1 axes, times a bivariate exponential, see Eq. (5.22) and the left-hand side of Fig. 5.13. It is a spiky function that fits the usual behavior of DFD; it is but a small variation of the piecewise function we introduced in [130] for coding motion-compensated prediction errors:

$$\phi_1(x_1, x_2) = \phi_{bs}(x_1)e^{-(x_1^2+x_2^2)}, \quad (5.22)$$

where ϕ_{bs} is

$$\phi_{bs}(x) = \begin{cases} -2 + 3|x| & \text{if } 0 \leq |x| < 1 \\ 2 - |x| & \text{if } 1 \leq |x| < 2 \\ 0 & \text{if } |x| \geq 2 \end{cases}. \quad (5.23)$$

- A Gabor function with oscillations in both the x_1 and the x_2 directions and with a frequency independent of the scaling factors (see Fig. 5.13). Note that this function has an additional parameter for the frequency but has only two possible rotations that correspond to the vertical and horizontal positions:

$$\phi_2(x_1, x_2) = \cos(\omega_x x) \cos(\omega_y y) e^{-(x_1^2+x_2^2)}. \quad (5.24)$$

In our implementation, we set $\omega_x = \omega_y$, but remark that the rotations are still useful because of the anisotropic scaling.

- A simple rectangular function expressed by Eq. (5.25), able to code errors due to the block-based nature of the motion compensation:

$$\phi_3(x_1, x_2) = \begin{cases} 1 & \text{if } |x_1| < 1 \wedge |x_2| < 1 \\ 0 & \text{otherwise} \end{cases}. \quad (5.25)$$

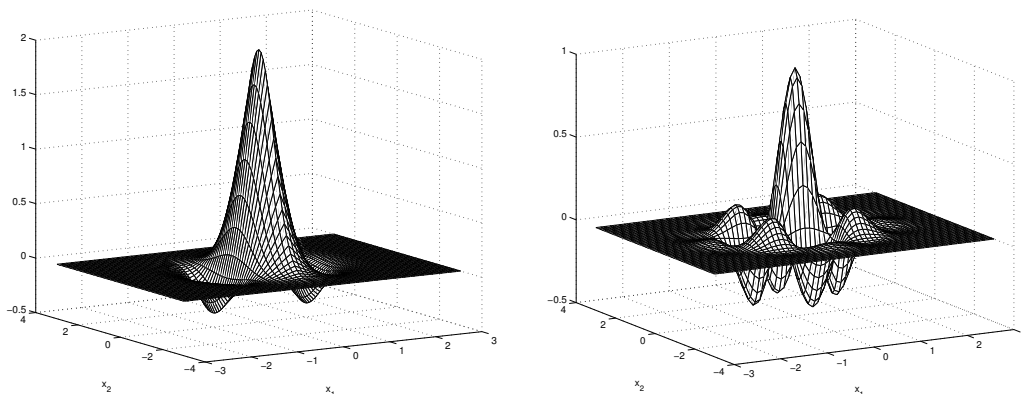


Figure 5.13: Generating functions ϕ_1 and ϕ_2

Note that this generating function, like the previous one and unlike the second derivative of a B-Spline, has a reduced set of possible rotations since the only two orientations we are interested in are vertical and horizontal.

The whole dictionary is composed by 2-D atoms whose spatial support is limited since, where the normalized atom has a value smaller than a certain threshold, it is set to zero. It is important to observe that, given a very small threshold, this choice does not affect at all the quality of the decomposition but, on the other hand, reduces the computational time.

Taking into account all atom parameters and the three generating functions, the dictionary can be written as:

$$\mathcal{D} = \{g_{(\phi, \vec{b}, \theta, \omega, \vec{a})}^{\text{norm}}(\vec{x})\}_{\phi, \vec{b}, \theta, \omega, \vec{a}}. \quad (5.26)$$

Here the index ϕ specifies which function has been chosen to create the atom, ω is the frequency, used only for the Gabor functions, while the other values are the same as in (5.21). Finally the number of waveforms in our dictionary (the parameter s in Eq. (4.25)) is approximately 1000: each of them can additionally be translated in any location of the image (see Eq. (5.15)). This set of atoms proves to be highly redundant.

5.2.4 Searching Algorithm

Matching Pursuit decomposes a DFD into its most important features. As was observed for still pictures in Section 5.1.2, since the dictionary is composed of all the translations of the transformed generating functions, all the inner products between the TGF translated all over the residual and the residual itself, correspond to the convolution of the TGF with the residual.

In order to speed up the search, convolutions are computed as products in the frequency domain, as done for still images and as depicted in Fig. 5.14. The Fourier transform of the entire dictionary is computed only once at the beginning of the video sequence and stored. Direct and inverse Fourier transforms are computed in a fast way using the FFTW package (<http://www.fftw.org>) (version 3.0.1, see [72]).

Even with this method the atom selection is still too slow for our purposes. Here we propose two solutions to speed up the algorithm. The first method is the Multiple Atom MP algorithm described in Section 4.4.3. It consists into a slightly modified version of MP: at each iteration more than one

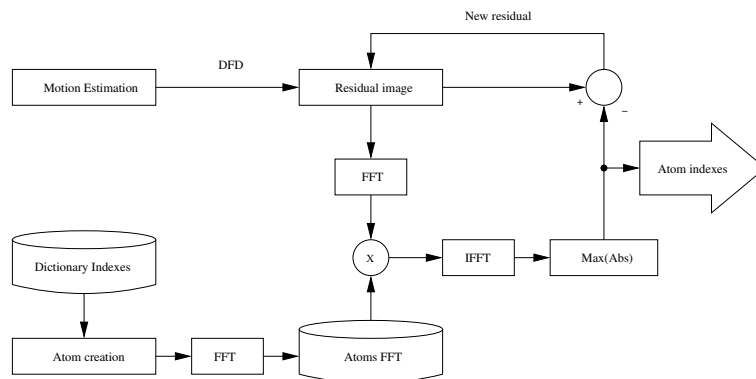


Figure 5.14: Scheme for the atom selection in the Fourier domain

atom is selected and used to decompose the residual. This can be done since in an image there are structures that are definitely separated in the spatial domain, and this is even more evident in a DFD where the features to code are usually small. Selecting on average \bar{k}_j atoms at once it turns out that MP only needs m/\bar{k}_j iterations, reducing in this way the number of inner products which constitute the most computationally demanding part of the algorithm. For example, decomposing a QCIF sequence, we observed a speed-up factor of around 10. The drawback of this method is that there is no more guaranty that at each iteration the best atom will be selected as in the case of the full search MP. However, the resulting loss in image quality is almost negligible.

A second possible strategy to speed up the searching algorithm can be found considering that from one iteration to another usually only a small area of the residual image changes. At the first iteration, all the convolutions between the image and each atom are computed; the main idea of this method is to store these values and at the next iteration update them only in the region where the best atom has been placed. The gain lays in performing the convolution and the inverse Fourier transform on a smaller area. The gain increases as selected atoms get smaller (have a smaller surface). This solution is possible only because the atoms in use have a limited spatial support. Such method has no quality loss and, according to our simulations, gives a gain in computational time of around 20% compared with the full search in the Fourier domain [116]. On the other hand, the required memory increases by around 30%.

The two presented algorithms permit to speed up the atom selection procedure, but unfortunately they are not compatible. The “multiple atom” search gives a higher reduction in terms of computational load and therefore is perhaps the most useful. However the second method is still interesting since it turns out to be completely lossless with respect to the full search.

5.2.5 Quantization and Entropy Coding

As said above, the parameters that specify an atom in the dictionary are the generating function type, two scale factors, the rotation angle and, only for Gabor atoms, the frequency. Moreover, we have to add to this list the atom position (two natural numbers whose range is determined by the frame size) and its projection coefficient. The indexes that characterize the atom shape are entropy coded using an adaptive arithmetic coding algorithm. Exactly as in the case of still pictures, since the rotation depends on the x_2 -scale, the arithmetic algorithm uses the conditioned probability $p(\text{rotation}|x_2\text{-scale})$ to code the rotation parameter.

In order to code the positions and projections of the atoms, two different approaches can be

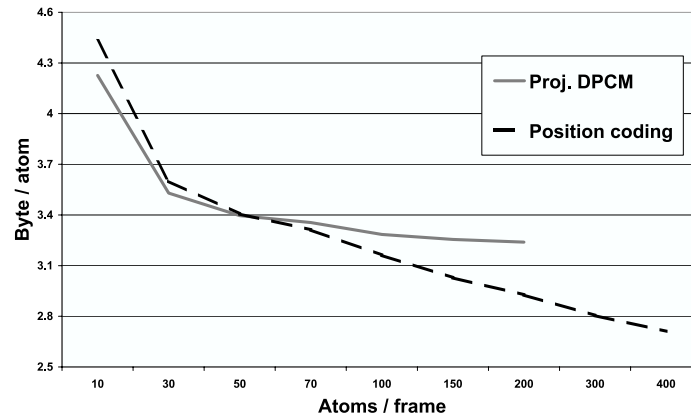


Figure 5.15: Bytes per atom necessary to code 19 frames of *container* QCIF using different encoding styles

taken into account. The first one consists in ordering the atoms according to their decreasing projection absolute values, then the projections are quantized in a differential way (DPCM) followed by arithmetic coding; the x_1 and x_2 coordinates are simply stored without any particular coding scheme. We will refer to this scheme as “projection DPCM” coding. The second approach performs a different sorting of the atoms in such a way to take advantage in coding the atoms positions [133], coding the coordinates in a differential way followed by arithmetic coding. We will refer to this scheme as “position” coding. Another interesting approach for coding the atoms is presented in [114], where bit-plane quantization of atom projections and quadtree prediction of atom positions are combined.

For both “projection DPCM” and “position” coding, quantization is performed in-loop: this provokes the re-injection of quantization error in the coding loop and permits to encode such error. For a detailed study about in-loop quantization for MP we recommend [45, 46]. Yet, we have to emphasize that our approach is independent and does not follow the modelization that is proposed in the cited paper.

“Position” vs. “Projection” Coding

Empirically, at very low bit-rates, when just few atoms per frame are coded, the projection DPCM method gives the best results. When the number of atoms per frame increases, the position encoding improves and finally outperforms the projection DPCM; later, the gap between these two coding styles increases together with the number of atoms selected (see Fig. 5.15). This phenomenon is easily explicable, since the position DPCM performances are related to the atoms density in the frame.

For example, simulations showed that for QCIF sequences usually the switching point is around 50 atoms/frame, after this threshold position encoding starts to outperform projection DPCM. With 200 atoms/frame the average gain is around 10% of the rate [116]. Fig. 5.16 shows the percentage of bits allocated to code the atoms parameters, positions and projections in both cases.

Remark that this situation is different from the one observed for still pictures where “position

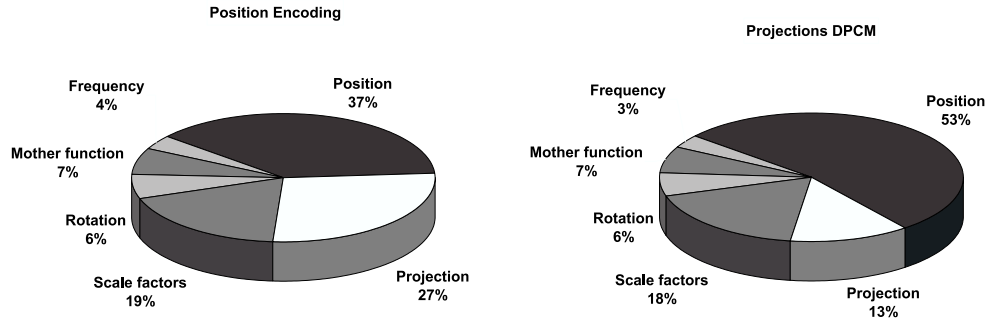


Figure 5.16: Example of typical bit allocations for “position” and “projection DPCM” encoding styles

coding” turns out to be the best solution at any bit-rate, as shown in the previous section.

An adaptive solution

The situation illustrated in Fig. 5.15 suggests that we can optimize the coding procedure by running both the previously illustrated entropy encoders and choosing the best one. In practice, after the position coding has been selected for few consecutive iterations we can stop checking and start to use this method only. In this way we always adopt the best coding solution, and from a rate point of view the only price to be payed is absolutely negligible: one bit per frame to specify the coding style. The possibility to switch from one encoding method to the other is integrated in the rate-distortion optimization, explained in next section.

5.2.6 Rate-Distortion Optimization

In a video sequence some consecutive frames are very similar one to each other: in this case the DFD contains very few information and, in our MP implementation, it can be coded with a small number of atoms. On the other hand, there are situations in which the amount of information to code strongly increases, requiring more atoms. Hence, given a certain target bit-rate, or a fixed quality, we have to face the problem of choosing the number of atoms per frame. A classical approach to this kind of issues is based on the minimization of a Lagrangian rate-distortion functional [143]:

$$\min\{J = D + \lambda R\}, \quad \text{with } \lambda \geq 0. \quad (5.27)$$

In Eq. (5.27), D is the distortion (MSE) and R is the rate (byte/second); λ is constant for the whole sequence. For a convex problem, the necessary and sufficient condition to find the absolute minimum of J is:

$$\frac{\partial D}{\partial m} = -\lambda \frac{\partial R}{\partial m}. \quad (5.28)$$

The first term in (5.28) is the variation of MSE through iterations, a negative number whose value is linked to the energy of the residual that an atom is able to catch. The second term represents the weighted differential rate. We can state that $\frac{\partial R}{\partial m}$ is always positive and in average decreases with n . Hence $-\lambda \frac{\partial R}{\partial m}$ is negative and increases. In order to minimize J we need a last consideration: the two

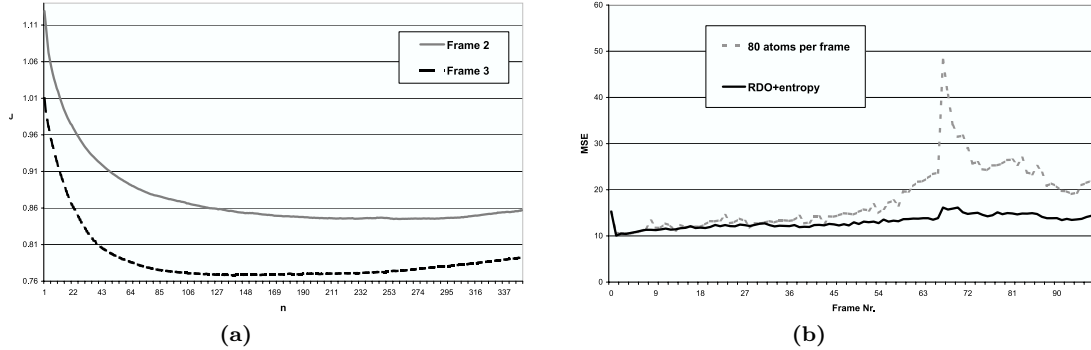


Figure 5.17: Left: rate-distortion optimization: $J(m)$ for two frames of *Stefan*. Right: MSE for the first 100 frames of *news* coded with and without RD optimization

terms of Eq. (5.28) are both negative and they increase in average with decreasing first derivative, but their limit when $n \rightarrow \infty$ is different (the first limit comes from Lemma 2 in [120]):

$$\lim_{m \rightarrow \infty} \frac{\partial D}{\partial m} = 0 \text{ and } \lim_{m \rightarrow \infty} -\lambda \frac{\partial R}{\partial m} = C. \quad (5.29)$$

Let assume that the constant C is negative. Now we can have two cases: either

$$\lim_{m \rightarrow 0} \frac{\partial D}{\partial m} < \lim_{m \rightarrow 0} -\lambda \frac{\partial R}{\partial m}, \quad (5.30)$$

and it means that we do not have to code any atom, or

$$\lim_{m \rightarrow 0} \frac{\partial D}{\partial m} \geq \lim_{m \rightarrow 0} -\lambda \frac{\partial R}{\partial m}, \quad (5.31)$$

and we have to stop the expansion when the condition in (5.28) is respected. From Eq. (5.29), thanks to the continuity of the first derivative of R and D , and assuming that both $\frac{\partial R}{\partial m}$ and $-\lambda \frac{\partial R}{\partial m}$ with their first derivatives are monotonically decreasing (and not only in average), it comes that it exists only one point \tilde{m} which solves Eq. (5.28) and this point is the absolute minimum we are looking for. In theory, since the dictionary is finite, the constant C in (5.29) can assume the value 0, depending solution adopted for coding the atoms. Anyway this situation has no practical interest since we never use a number of atoms which can be comparable with the size of the dictionary.

From an implementation point of view, we have the problem that the differential MSE has a monotone trend but it does not always increase with m . The same observation holds for the differential rate. These small deviations from the ideal behavior imply the possible existence of local minima. However this problem can be easily solved, since $J(m)$ always shows a precise trend, as can be seen in Fig. 5.17(a). The only precaution we take is not to stop the coding process exactly when J starts to increase, but to go on for few iterations in order to be sure that we are not in a local minimum.

Concluding, given a required quality factor, the master coder fixes the value of the parameter λ . An amount of bits is then assigned to each frame according to the rate control of the master coder.

It is important to point out that this RD approach can be used even when the atom selection is performed by turning to the “multiple atom” algorithm (see subsection 5.2.4). In this case, however,

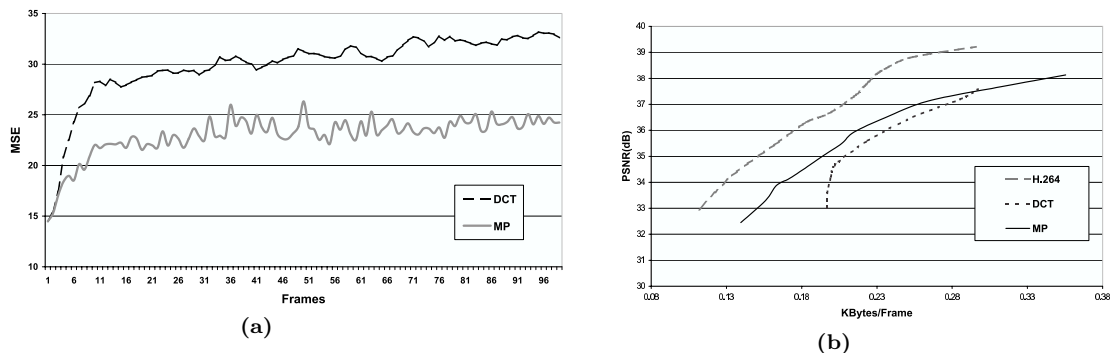


Figure 5.18: Left: MSE obtained coding the first 100 frames of *container* using MP (0.190 KBytes/frame) and the 8x8 DCT (0.194 KBytes/frame); no I-frames. Right: RD curves obtained coding the first 100 frames of *traffic* using 8x8 DCT and MP with the same motion estimation and H.264. I-frames enabled

some changes are required, due to the fact that atoms are not necessarily selected in decreasing order of projection absolute value. Hence at the first step we subtract all the selected atoms from the residual but we code only the best one, and we put all the others in a list sorted by decreasing projections. In the following steps we code the best of the current step plus all the atoms in the list whose projection is higher than the projection of the best atom of the current step.

In order to compute the rate, two different situations have to be taken into account since we do not know a-priori if a position or projection DPCM coding style will be adopted (see section 5.2.5). Also the choice between these methods is then left to the RD algorithm.

Figure 5.17(b) shows the MSE behavior of the test sequence *news*. It is easy to observe the improvement achieved by the RD optimization with respect to the case in which a fixed number of atom per frame is coded.

5.2.7 Results and Comparisons

The first comparisons are aimed at testing the quality of the MP codec with respect to a standard 8x8 DCT. So we adopt the same motion estimation described in Section 5.2.1 and we code then the DFDs using either MP or a classical DCT block-based scheme. The MP atom selection is performed using the fast multiple atom algorithm, explained in Section 5.2.4. In this case, for all the tested sequences the MP outperforms DCT. For example Fig. 5.18(a) shows the MSE behavior for the sequence *container* in QCIF format: even if the DCT has a slightly higher rate, it is outperformed by MP in terms of both visual quality and mean square error.

In Fig. 5.18(b) one can see the RD curve obtained by coding a video-surveillance traffic sequence (QCIF format), allowing the encoders to put I-frames when necessary. Comparisons show the superiority of MP versus DCT, especially at very low bit-rates. Figure 5.19 shows a frame of the sequence *container* in QCIF format (176×144 pixels) compressed with the proposed method based on MP (on the left) and 8x8 DCT (right). The motion compensation used to obtain the DFD is, of course, the same. The quality of the video obtained with the proposed method is higher, even if the single picture in the figure might not be able to show it out properly.

Moreover, for a video codec time constraints are critical, especially at the decoder side. Matching



Figure 5.19: 16th frame of the sequence *container* in QCIF format coded with MP (left) and 8x8 DCT (right)

Pursuit is an asymmetric algorithm in the sense that the complexity of the encoder is much more higher than the one of the decoder. Indeed, while the selection of the atoms approximating a signal has to follow the procedure explained in Section 4.4, the reconstruction just implies a linear combination of the functions corresponding to the received indexes. A reduced redundancy factor of the dictionary with respect to one used for still pictures together with the fact that the spatial support of the atoms is limited by setting to zero their value when it is below a certain threshold help in moderating the complexity. These factors have to be added to the use of a fast algorithm and of the fast Fourier transform for the atom selection. Thanks also to others technical minor algorithm optimizations [116] a real time decoding is possible for sequences up to CIF format (352×288 pixels).

In order to compare the MP video coder with H.264 we disabled some of the options not yet implemented in our motion estimation. Following settings have been used:

- Hadamar transform: enabled,
- Search range: 16,
- Number of reference frames: 1,
- Block sizes (for motion estimation): all enabled,
- B frames: disabled,
- CABAC: disabled.

Results clearly show that H.264 obtains better performances than our encoder. For example coding the sequence *traffic* in QCIF format we can observe a gap of more than 1.5 dB (see Fig. 5.18(b)). This gap can be explained assuming that the H.264 encoder is fully optimized for the block-based integer transform, while we work in a frame-based way. In fact we notice that, especially at low bit-rates, the losses due to a coding syntax not suited for the overall coder heavily affect the performances of MP. We also have to consider that, even with some disabled option, the motion estimation of H.264 is still more accurate than the one we used in our MP implementation (see also Section 5.2.1). In fact we did not disable all the features missing in our MC algorithm and this results in a not completely fair comparison between the two approaches.

5.3 Discussion

In this chapter we have seen two methods of coding for still pictures and videos. Both are based on the use of a very redundant geometric-oriented dictionary and the function selection is entrusted to a greedy algorithm. The promising results obtained by these examples confirm of the considerations made in the previous chapters. In addition, the fact that the compression quality obtained is good in spite of the huge size of the dictionary shows the pertinence of the representation techniques.

It is worth emphasizing that the families of basis function composing the dictionary, together with decomposition by means of greedy algorithms can offer the advantage of generating progressive stream. This is a key feature in the design of adaptive visual communication applications, where rate scalability (also known as SNR scalability) in general is becoming an important requirement. This functionality is not directly implemented in the encoders we have presented here, but it is strongly connected to the representation method we use. Therefore it can be achieved by minor modifications. An example of a coding scheme where rate and spatial scalabilities are effectively present is given in [74, 75].

Only results concerning gray-scale pictures are presented. The reason is that we are interested more in the structure of the images than in their color components. Nevertheless the extension to the chrominance components is feasible. Let us just cite [70], where a color image coder based on a MP expansion is studied.

Chapter 6

On the use of *a priori* information for sparse signal decomposition

Throughout this dissertation we have been looking for an efficient representation or approximation of a signal by means of a linear expansion into a possibly overcomplete family of functions. In Chapter 4 we have seen that efficiency is often characterized by sparseness and that, in general, the problem of recovering the sparsest signal approximation (or representation) over a redundant dictionary is an NP-hard problem. We have also seen that there exist several faster methods, which can solve this problem, provided that the dictionary respects certain conditions. Such recovery conditions can be roughly summarized by the assumption of quasi-incoherence.

However, experience and intuition dictate that good dictionaries for approximations of natural signals can be very redundant and, depending on the kind of structures we want to describe, they may be highly coherent. We can observe a strong gap between these characteristics and the theoretical recovery conditions stated in Chapter 4, asking for a quasi-incoherent dictionary. How can we handle this discrepancy between theory and practice? The question of achieving sparseness in some class of coherent dictionaries is already faced in [78], but for the particular case of Vandermonde and Fourier dictionaries. The solution that we propose and discuss here is based on the potentiality of using *a priori* knowledge in the atom selection procedure. Intuitively, a huge amount of information is available about the natural signals we are dealing with. This can come from the knowledge of its physical background as well as from an empirical study. Exploiting such *a priori* information can help us in finding better approximations and representations.

The main points we face in this chapter are how to represent the *a priori* information, with which algorithm it can be used and which improvements it can bring. Theoretical and practical sides of the problem are studied and in addition we provide several examples that show the achievable benefits and functional ways to compute reliable and useful *priors* about a signal. The first part of the chapter describes the Weighted Basis Pursuit Denoising and Weighted Matching Pursuit principles, studying their approximation properties. In the second part the exact representation case is briefly examined. Finally we give a hint about two applications of the presented framework and comprehensively discuss it.

6.1 Mathematical Setting

Let us firstly recall some notations and definitions that will be useful for the rest of the chapter. We analyze again the two problems introduced in Chapter 4:

- sparse m -term approximation:

$$\min_{\mathbf{c} \in \mathbb{R}^d} \|f - D\mathbf{c}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{c}\|_0 \leq m, \quad (6.1)$$

- exact sparse representation:

$$\min_{\mathbf{c} \in \mathbb{R}^d} \|\mathbf{c}\|_0 \quad \text{s.t.} \quad D\mathbf{c} = f. \quad (6.2)$$

In the previous expressions $f \in \mathcal{H}$ is the function to be analyzed and \mathcal{H} is a Hilbert space (unless otherwise stated, it is assumed that $\mathcal{H} \equiv \mathbb{R}^n$). The dictionary is defined as $\mathcal{D} = \{g_i : i \in \Omega\}$ where $\forall i \ \|g_i\|_2 = 1$ and $|\Omega| = d$. D is the $n \times d$ synthesis matrix of the dictionary, where each one of the columns corresponds to an atom. Finally, \mathbf{c} is the vector of coefficients to be recovered.

The *a priori* knowledge is expressed by the diagonal matrix $W(f, \mathcal{D})$, defined as follows:

Definition 6.1 *A weighting matrix $W = W(f, \mathcal{D})$ is a square diagonal matrix of size $d \times d$. Each of the entries $w_i \in (0, 1]$ from the diagonal corresponds to some measure of the *a priori* likelihood of a particular atom $g_i \in \mathcal{D}$ to be part of the sparsest decomposition of f .*

Notice that, given some f , all the atoms in the dictionary are assumed to have some non-zero *a priori* probability. In effect, those that would have a zero weight are considered to be excluded from the dictionary.

Consistently with the notation adopted in Chapter 4, Γ stands for the optimal subset of the index set Ω . Optimality has to be interpreted in the sense of (6.1) when speaking about approximation and of (6.2) when dealing with the exact representation problem. We also define $w_{\bar{\Gamma}}^{max}$ as the biggest weight corresponding to the subset of atoms indexed in $\bar{\Gamma} = \Omega \setminus \Gamma$, hence:

$$w_{\bar{\Gamma}}^{max} = \max_{\gamma \in \bar{\Gamma}} w_{\gamma}. \quad (6.3)$$

Moreover, an additional quantity is required in the results depicted below:

$$\epsilon_{max} = \max_{\gamma \in \Gamma} (1 - w_{\gamma}^2). \quad (6.4)$$

Eqs. (6.3) and (6.4) concern the goodness of the *a priori* information about the signal f (and thus they do depend on f even if not explicitly stated). The reader will notice that these quantities also depend on the optimal set of atoms Γ , preventing from establishing a rule to compute them in advance. The role of these magnitudes is to represent the influence of the *prior* in the results obtained below. Notice that $0 \leq \epsilon_{max} < 1$ and $0 < w_{\bar{\Gamma}}^{max} \leq 1$.

Definition 6.2 *ϵ_{max} is close to zero if “good” atoms (the ones belonging to Γ) are not penalized by the *a priori* information. In such a case we state that the *a priori* knowledge is “reliable”.*

The quantity $w_{\bar{\Gamma}}^{max}$ becomes small if all “bad” atoms are strongly penalized by the *a priori* knowledge. Notice that the “reliability” does not impose any condition on $w_{\bar{\Gamma}}^{max}$.

The weights are not arbitrary and are not supposed to be independently and blindly optimized by the algorithm during the subset selection procedure. These values alone are not meant to determine

whether an atom shall be included in the selection or not. They introduce a fuzzy likelihood that could be derived from a good parametric model on the interaction between signals and the dictionary.

The *a priori* matrix W allows a new signal dependent definition of the cumulative coherence $\mu_1(m)$. Indeed, the conditions that ensure the recoverability of the best m -term approximant rely on this quantity. Using *a priori* information, some atom interactions can be penalized or even dismissed in the cumulative coherence measure:

Definition 6.3 *The Weighted Cumulative Coherence function of \mathcal{D} is defined as the following data dependent measure:*

$$\mu_1^w(m, \mathcal{D}, W) = \sup_{|\Lambda|=m} \sup_{i \in \Omega \setminus \Lambda} \sum_{\lambda \in \Lambda} |\langle g_\lambda, g_i \rangle| \cdot w_\lambda \cdot w_i. \quad (6.5)$$

Note that if $W = I$, then $\mu_1^w(m, \mathcal{D}, I) = \mu_1(m, \mathcal{D})$. Moreover, $\forall m, \mathcal{D}, W$ we have that $\mu_1^w(m, \mathcal{D}, W) \leq \mu_1(m, \mathcal{D})$.

6.2 Approximation by Weighted Basis Pursuit Denoising

We investigate now the effects of inserting *a priori* knowledge in the convex relaxation of the subset selection problem. In Section 4.5.3 we have seen a Bayesian formulation of Basis Pursuit Denoising, observing how the most probable coefficients for representing a signal f can be found as (see Eq. (4.37)):

$$\mathbf{b}_P = \arg \max_{\mathbf{b}} \left(\ln(p(f|D, \mathbf{b})) + \sum_i \ln p(b_i) \right) = \arg \min_{\mathbf{b}} \left(\frac{\|f - D\mathbf{b}\|_2^2}{2\sigma_r^2} + \sum_i \frac{\sqrt{2}|b_i|}{\sigma_i} \right).$$

Making the hypothesis that the variance of b_i (σ_i^2) is constant for every index i means that the most probable coefficient vector \mathbf{b} is the one found by the BPDN principle. But this hypothesis does not often correspond to reality. On the contrary, if the variances of the coefficients are not forced to be all the same, it turns out that the most probable signal representation can be found by solving the following problem:

$$(P_{2-1}^w) \quad \min_{\mathbf{b} \in \mathbb{R}^d} \frac{1}{2} \|f - D\mathbf{b}\|_2^2 + \gamma \|W^{-1}\mathbf{b}\|_1, \quad (6.6)$$

where the diagonal matrix with entries in $(0, 1]$ is defined in Section 6.1. One can notice that in Eq. (6.6), the introduction of weights allows to individually model the components of \mathbf{b} . Since this approach introduces *a priori* information under the form of weights in the BPDN paradigm, therefore from now on, we will refer to (P_{2-1}^w) as Weighted Basis Pursuit Denoising or WBPDN.

Let us now study the relationship between the results obtained by solving problem (P_{2-1}^w) and (P_{2-0}) . In general we refer with \mathbf{b} to the coefficient vector of the relaxed problem and with \mathbf{c} to the one of the problem (P_{2-0}) :

$$(P_{2-0}) \quad \min_{\mathbf{c} \in \mathbb{R}^d} \|f - D\mathbf{c}\|_2^2 + \tau^2 \|\mathbf{c}\|_0.$$

Note that, given an arbitrary index subset $\Lambda \subset \Omega$, in the following \mathbf{c}_Λ and \mathbf{b}_Λ lay in \mathbb{R}^Λ but sometimes these are extended to \mathbb{R}^Ω by padding with zeros. The same is valid for the matrix W_Λ . First, let us introduce the Weighted Recovery Factor:

Definition 6.4 *Given a dictionary \mathcal{D} indexed in Ω and an index subset $\Lambda \subset \Omega$, we define the Weighted Recovery Factor (WRF) as:*

$$WRF(\Lambda) = \sup_{i \notin \Lambda} \left\| (D_\Lambda W_\Lambda)^+ g_i \cdot w_i \right\|_1. \quad (6.7)$$

6.2.1 Preliminary Propositions

Here, some preliminary propositions are presented, allowing us to prove the results of the following. Given $\Lambda \subset \Omega$, let us call f_Λ the approximant of f that uses the coefficients indexed in Λ , i.e. $f_\Lambda = DD_\Lambda^+ f$. Let us also call W_Λ the weighting matrix restricted to the indexes in Λ .

The next lemma, similarly to the ‘‘Correlation Condition Lemma’’ in [175], basically states that, if the atoms of Λ have a small correlation with the residual $(f - f_\Lambda)$, then the support of any vector that solves (P_{2-1}^w) is a subset of Λ . This result will be used to prove Theorem 6.1.

Lemma 6.1 *Given an index subset $\Lambda \subset \Omega$, suppose that the following condition is satisfied:*

$$\|D^T(f - f_\Lambda)\|_\infty < \frac{\gamma}{w_\Lambda^{max}} \cdot (1 - WRF(\Lambda)), \quad (6.8)$$

where $w_\Lambda^{max} \in (0, 1]$ is the quantity defined by equation (6.3). Then, any coefficient vector \mathbf{b}_* that minimizes the cost function of problem (P_{2-1}^w) must have a support contained in Λ .

Proof: Assume that \mathbf{b}_* minimizes (6.6), but it uses an index outside Λ . One can compare \mathbf{b}_* with its projection $D_\Lambda^+ D\mathbf{b}_*$, which is supported in Λ , obtaining:

$$2\gamma (\|W^{-1}\mathbf{b}_*\|_1 - \|W_\Lambda^{-1}(D_\Lambda^+ D\mathbf{b}_*)\|_1) \leq \|f - DD_\Lambda^+ D\mathbf{b}_*\|_2^2 - \|f - D\mathbf{b}_*\|_2^2. \quad (6.9)$$

First, we shall provide a lower bound on the left-hand side of the previous inequality. Let us split the vector \mathbf{b}_* into two parts: $\mathbf{b}_* = \mathbf{b}_\Lambda + \mathbf{b}_{\bar{\Lambda}}$, where the former vector contains the components with indexes in Λ , while the latter the remaining components from $\bar{\Lambda} = \Omega \setminus \Lambda$. Acting as in the proof of the Correlation Condition Lemma in [175] it follows that:

$$\|W^{-1}\mathbf{b}_*\|_1 - \|W_\Lambda^{-1}(D_\Lambda^+ D\mathbf{b}_*)\|_1 \geq (1 - WRF(\Lambda)) \cdot \|W^{-1}\mathbf{b}_{\bar{\Lambda}}\|_1. \quad (6.10)$$

For more details, see [50]. The quantity appearing on the right-hand side of (6.9) does not depend on the weighting matrix, thus, exactly as in [175], it can be upper bounded by $2\|\mathbf{b}_{\bar{\Lambda}}\|_1 \cdot \|D^T(f - f_\Lambda)\|_\infty$. This, together with (6.9) and (6.10), gives:

$$\gamma(1 - WRF(\Lambda)) \cdot \|W^{-1}\mathbf{b}_{\bar{\Lambda}}\|_1 \leq \|\mathbf{b}_{\bar{\Lambda}}\|_1 \cdot \|D^T(f - f_\Lambda)\|_\infty. \quad (6.11)$$

Since the weights are in $(0, 1]$, and the vector $\mathbf{b}_{\bar{\Lambda}}$, by assumption, cannot be null, it can be written:

$$\gamma(1 - WRF(\Lambda)) \leq \frac{\|\mathbf{b}_{\bar{\Lambda}}\|_1}{\|W^{-1}\mathbf{b}_{\bar{\Lambda}}\|_1} \cdot \|D^T(f - f_\Lambda)\|_\infty \leq w_\Lambda^{max} \cdot \|D^T(f - f_\Lambda)\|_\infty. \quad (6.12)$$

If (6.8) is valid, then (6.12) fails and so one must discard the hypothesis that \mathbf{b}_* is non-zero for an index in $\bar{\Lambda}$. ■

We now focus on finding a necessary and sufficient condition for the existence and uniqueness of a minimum of (P_{2-1}^w) . The presence of the ℓ_1 norm implies that the cost function of this problem is non-smooth at zero: for this reason the concept of *subdifferential* is used. Given a real vector variable \mathbf{x} , the subdifferential of $\|\mathbf{x}\|_1$ is denoted by $\partial\|\mathbf{x}\|_1$ and defined as:

$$\partial\|\mathbf{x}\|_1 = \{\mathbf{u} | \mathbf{u}^* \mathbf{x} = \|\mathbf{x}\|_1, \|\mathbf{u}\|_\infty \leq 1\}.$$

The vectors \mathbf{u} that compose the subdifferential are called *subgradients* [109].

Lemma 6.2 *A necessary and sufficient condition for \mathbf{b}_* to globally minimize the objective function of (P_{2-1}^w) over all coefficient vectors with support Λ is that:*

$$\mathbf{c}_\Lambda - \mathbf{b}_* = \gamma (D_\Lambda^T D_\Lambda)^{-1} W_\Lambda^{-1} \mathbf{u}, \quad (6.13)$$

where \mathbf{u} is a vector from $\partial \|\mathbf{b}_*\|_1$. Moreover, the minimizer is unique.

The proof of this lemma, technical and very similar to the proof made by Fuchs in [77], can be found in the Appendix.

If $W = I$, then this result coincides with the one developed by Fuchs in [77] and by Tropp in [175], in the complex case. Next lemma concludes the preliminary propositions bounding the error in the coefficients domain.

Lemma 6.3 *Suppose that \mathbf{b}_* minimizes the cost function of problem (P_{2-1}^w) . Then the following bound holds:*

$$\|\mathbf{c}_\Lambda - \mathbf{b}_*\|_\infty \leq \frac{\gamma}{w_\Lambda^{\min}} \cdot \left\| (D_\Lambda^T D_\Lambda)^{-1} \right\|_{\infty, \infty},$$

where w_Λ^{\min} is defined as

$$w_\Lambda^{\min} = \inf_{i \in \Lambda} w_i. \quad (6.14)$$

Proof: Let us consider the necessary and sufficient condition of Lemma 6.2: taking the ℓ_∞ norm of (6.13) we obtain:

$$\|\mathbf{c}_\Lambda - \mathbf{b}_*\|_\infty = \gamma \left\| (D_\Lambda^T D_\Lambda)^{-1} W_\Lambda^{-1} \mathbf{u} \right\|_\infty \leq \gamma \left\| (D_\Lambda^T D_\Lambda)^{-1} W_\Lambda^{-1} \right\|_{\infty, \infty} \cdot \|\mathbf{u}\|_\infty.$$

By definition of subdifferential, $\|\mathbf{u}\|_\infty \leq 1$. Inserting this into the previous equation and using the sub-multiplicative property of matrix norms ($\|AB\|_{p,q} \leq \|A\|_{p,q} \cdot \|B\|_{p,q}$), we can prove that

$$\|\mathbf{c}_\Lambda - \mathbf{b}_*\|_\infty \leq \gamma \left\| (D_\Lambda^T D_\Lambda)^{-1} \right\|_{\infty, \infty} \cdot \|W_\Lambda^{-1}\|_{\infty, \infty}.$$

Just apply the fact that $\|W_\Lambda^{-1}\|_{\infty, \infty} = \sup_{i \in \Lambda} (1/w_i) = \frac{1}{w_\Lambda^{\min}}$ to reach the result. ■

6.2.2 Recovery Conditions of WBPDN

Suppose now that \mathbf{c}_Γ is the sparsest solution to (P_{2-0}) and that its support is Γ , with $|\Gamma| = m$. D_Γ will be the matrix containing all the atoms participating to the sparsest approximation of f and f_m^{opt} will be the approximant given by \mathbf{c}_Γ , i.e. $f_m^{\text{opt}} = D\mathbf{c}_\Gamma = DD_\Gamma^\dagger f = D_\Gamma D_\Gamma^\dagger f$. Assuming $WRF(\Gamma) < 1$, we have the following result.

Theorem 6.1 *Given $\tau > 0$, trade-off parameter of the problem (P_{2-0}) , suppose that \mathbf{b}_* minimizes the cost function of problem (P_{2-1}^w) with threshold*

$$\gamma = \frac{\tau \cdot w_\Gamma^{\max}}{1 - WRF(\Gamma)}, \quad (6.15)$$

where w_Γ^{\max} is defined in (6.3). Then:

1. WBPDN never selects a non optimal atom since $\text{support}(\mathbf{b}_*) \subset \Gamma$.
2. The solution of WBPDN is unique.

3. The following upper bound is valid:

$$\|\mathbf{c}_\Gamma - \mathbf{b}_*\|_\infty \leq \frac{\tau \cdot \frac{w_\Gamma^{max}}{w_\Gamma^{min}} \cdot \left\| (D_\Gamma^T D_\Gamma)^{-1} \right\|_{\infty, \infty}}{1 - WRF(\Gamma)}. \quad (6.16)$$

4. The support of \mathbf{b}_* contains every index j for which

$$|c_\Gamma(j)| > \frac{\tau \cdot \frac{w_\Gamma^{max}}{w_\Gamma^{min}} \cdot \left\| (D_\Gamma^T D_\Gamma)^{-1} \right\|_{\infty, \infty}}{1 - WRF(\Gamma)}. \quad (6.17)$$

The scalar w_Γ^{min} appearing in Eqs. (6.16) and (6.17) is defined in (6.14).

Proof: Considering the first stated result, note that every atom indexed by Γ has zero inner product with the optimal residual ($r_m^{opt} = f - f_m^{opt}$) since f_m^{opt} is the best approximation of f using the atoms in Γ . Using Proposition 5.1 in [175] and recalling that \mathcal{D} is finite, it can be stated that

$$\|D^T(f - f_m^{opt})\|_\infty < \tau. \quad (6.18)$$

Moreover, Lemma 6.1 guarantees that for any γ satisfying

$$\|D^T(f - f_m^{opt})\|_\infty < \frac{\gamma}{w_\Gamma^{max}} \cdot (1 - WRF(\Gamma)), \quad (6.19)$$

the solution \mathbf{b}_* to the convex problem (P_{2-1}^w) is supported on Γ . From (6.18) and (6.19) it follows that for any γ that satisfies the following condition, it is insured that $support(\mathbf{b}_*) \subset \Gamma$:

$$\gamma \geq \frac{\tau \cdot w_\Gamma^{max}}{1 - WRF(\Gamma)}. \quad (6.20)$$

In the following, the smallest possible value for γ is chosen, so that, Eq. (6.20) becomes an equality. The uniqueness of the solution follows from Lemma 6.2. With regard to the third point, Lemma 6.3 yields

$$\|\mathbf{c}_\Gamma - \mathbf{b}_*\|_\infty \leq \frac{\gamma}{w_\Gamma^{min}} \left\| (D_\Gamma^T D_\Gamma)^{-1} \right\|_{\infty, \infty} \leq \frac{\tau \cdot \frac{w_\Gamma^{max}}{w_\Gamma^{min}}}{1 - WRF(\Gamma)} \cdot \left\| (D_\Gamma^T D_\Gamma)^{-1} \right\|_{\infty, \infty}.$$

Using Equation (6.20), the fourth result of the theorem can be proved exactly as in [175]. ■

This theorem states two important concepts. First, if the trade-off parameter is correct and the weighted cumulative coherence of the dictionary is small enough, WBPDN is able to select the correct atoms to obtain the sparsest signal approximation. Furthermore, the error made by the algorithm to compute the coefficients with respect to the optimal ones is bounded. The quantities w_Γ^{min} and w_Γ^{max} depend on the reliability and goodness of the *a priori* respectively. In particular, if W tends to be optimal (i.e. its diagonal entries tend to 1 for the elements that should appear in the sparsest approximation and to 0 for the ones that should not), $w_\Gamma^{min} \rightarrow 1$ and $w_\Gamma^{max} \rightarrow 0$. This results in an improved bound for the error of the coefficients and a condition for γ in Eq. (6.15) that is easier to respect. The reader will notice that it is quite improbable that such an “optimal” W exist in practice. Indeed, the typical information supplied by an *a priori* model will be quite imprecise (this, however, does not prevent a *prior* of being reliable and helpful). This aspect is discussed and justified at the end of Section 6.2.3. Compare these results with Theorem 4.4 in order to see the improvements brought by the use of *a priori* information. Indeed, if $W = I$, Theorem 6.1 boils down to Theorem 4.4.

Note that, once the algorithm has recovered the atom subset, the appropriate amplitudes of the coefficients can be computed by the orthogonal projection of the signal onto the space generated by the selected atoms. Hence, the error made by the algorithm in the coefficients computation is avoided (see Eq. (6.16)). This method is illustrated in Sections 4.5.5 and 6.2.4.

6.2.3 Relation with the Weighted Cumulative Coherence

In this subsection, previous results are described using the weighted cumulative coherence function defined in (6.5). In this way a comparison is made between the results achievable by BPDN and WBPDN.

Theorem 6.2 *Assume that the real vector \mathbf{b}_* solves (P_{2-1}^w) with*

$$\gamma = \frac{w_{\Gamma}^{max} \cdot \tau(1 - \epsilon_{max} - \mu_1^w(m-1))}{1 - \epsilon_{max} - \mu_1^w(m) - \mu_1^w(m-1)}.$$

Then $\text{support}(\mathbf{b}_) \subset \Gamma$ and*

$$\|\mathbf{b}_* - \mathbf{c}_{\Gamma}\|_{\infty} \leq \frac{\tau \cdot \frac{w_{\Gamma}^{max}}{w_{\Gamma}^{min}}(1 - \epsilon_{max} - \mu_1^w(m-1))}{(1 - \epsilon_{max} - \mu_1^w(m) - \mu_1^w(m-1))(1 - \mu_1(m-1))}. \quad (6.21)$$

Proof: This result can be obtained from [175] and Theorem 6.1, since:

$$\|\mathbf{b}_* - \mathbf{c}_{\Gamma}\|_{\infty} \leq \frac{\gamma}{w_{\Gamma}^{min}} \left\| (D_{\Gamma}^T D_{\Gamma})^{-1} \right\|_{\infty, \infty} = \frac{\tau \cdot \frac{w_{\Gamma}^{max}}{w_{\Gamma}^{min}}(1 - \epsilon_{max} - \mu_1^w(m-1)) \cdot \left\| (D_{\Gamma}^T D_{\Gamma})^{-1} \right\|_{\infty, \infty}}{(1 - \epsilon_{max} - \mu_1^w(m) - \mu_1^w(m-1))}.$$

Considering that

$$\left\| (D_{\Gamma}^T D_{\Gamma})^{-1} \right\|_{\infty, \infty} = \left\| (D_{\Gamma}^T D_{\Gamma})^{-1} \right\|_{1,1} \leq \frac{1}{1 - \mu_1(m-1)},$$

(see [77, 175]) proves equation (6.21). ■

This result illustrates how the distance between the optimal coefficients and the solution found by solving (P_{2-1}^w) can be bounded. In case no *prior* is given, the bound on the coefficient error is obtained from Eq. (6.21) setting $W = I$. Consequently, $w_{\Gamma}^{min} = 1$, $\epsilon_{max} = 0$ and $w_{\Gamma}^{max} = 1$, and we obtain the very same result as in Corollary 4.1:

$$\|\mathbf{b}_* - \mathbf{c}_{\Gamma}\|_{\infty} \leq \frac{\tau}{1 - \mu_1(m) - \mu_1(m-1)}. \quad (6.22)$$

Comparing the two bounds, one can observe how the availability of a reliable *prior* on the signal can help in finding a sparser signal approximation. This concept is emphasized in the following corollary.

Corollary 6.1 *Let $W(f, \mathcal{D})$ be a reliable a priori knowledge, with $w_{\Gamma}^{max}/w_{\Gamma}^{min} \leq 1$. Then for any positive integer m such that*

$$\mu_1^w(m-1) + \mu_1^w(m) + \epsilon_{max} < \mu_1(m-1) + \mu_1(m) < 1,$$

the error $\|\mathbf{b}_ - \mathbf{c}_{\Gamma}\|_{\infty}$ given by the coefficients found by WBPDN is smaller than the one obtained by BPDN. Hence, the bound stated by Eq. (6.21) is lower than the one in Eq. (6.22), i.e.*

$$\frac{\tau \cdot \frac{w_{\Gamma}^{max}}{w_{\Gamma}^{min}}(1 - \epsilon_{max} - \mu_1^w(m-1))}{(1 - \epsilon_{max} - \mu_1^w(m) - \mu_1^w(m-1))(1 - \mu_1(m-1))} \leq \frac{\tau}{1 - \mu_1(m) - \mu_1(m-1)}. \quad (6.23)$$

This result is proved in the appendix.

The reader may notice that if $\frac{w_{\Gamma}^{max}}{w_{\Gamma}^{min}} < 1$ the *a priori* information already tells which is the right support of the solution. Indeed, a simple threshold on the weights would find the appropriate set of atoms. This is an unrealistic situation in practice. However, provided that the *a priori* information is reliable, we do not need $\frac{w_{\Gamma}^{max}}{w_{\Gamma}^{min}} < 1$ to justify an improvement on the behavior of the algorithm. Suppose that the weights do not penalize the optimal atoms, but only some (not all) of the “wrong” ones: in this case $\frac{w_{\Gamma}^{max}}{w_{\Gamma}^{min}} = 1$. In such a situation, given that $\mu_1^w(m-1) + \mu_1^w(m) + \epsilon_{max} < \mu_1(m-1) + \mu_1(m) < 1$, Eq. (6.23) is still valid. This means that, even if the *a priori* knowledge is imprecise (but reliable), WBPDN can behave significantly better than BPDN. The same consideration applies to Eqs. (6.16) and (6.17).

6.2.4 Example: Use of Footprints and WBPDN for Sparse Approximation

It is time to give an example of approximation using *a priori* information; we thus consider the case where a piecewise-smooth signal is decomposed over an overcomplete dictionary. The dictionary is built by the union of an orthonormal basis defined by the *Symmlet-4* family of wavelets [118] and the respective family of footprints for all the possible translations of the Heaviside function (see [64]). The former is intended to represent the smooth part of the signal, while the latter is used to model the discontinuities. Footprints are functions composed by the superposition of all wavelet coefficients that a given deterministic singularity model (translations of the Heaviside function in our case) generates on a wavelet basis (see Fig. 6.1). The graphical representation of the dictionary matrix can be seen in Fig. 6.2, where the columns are the waveforms that compose the dictionary.

Such a dictionary is far from satisfying the sufficient conditions required to ensure the recovery of an optimal approximant with more than one term. Moreover, even if the best *a priori* was available, it is also far from satisfying the sufficient condition based on the weighted cumulative coherence. Nevertheless, we consider this example because of two main reasons. The first concerns the fact that sufficient theoretical conditions exposed in the literature are very pessimistic and reflect the worst possible case. The second reason is that, as previously discussed, experience seems to teach us that good dictionaries for efficient approximation of signals, are likely to be highly coherent. This fact conflicts with the requirement of incoherence for the good behavior of greedy algorithms. Hence, we find this example of special interest to underline the benefits of using *a priori* information and additional signal modeling for nonlinear expansions.

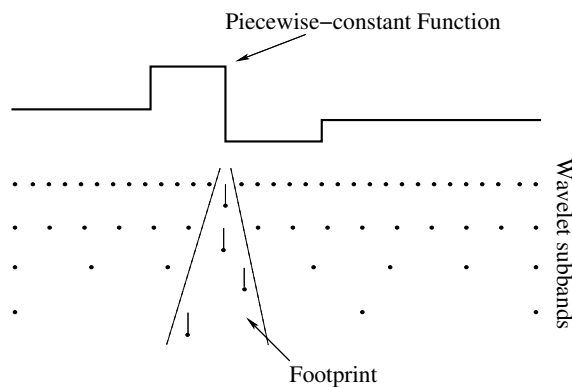


Figure 6.1: Wavelet Footprints description scheme for a piecewise-constant signal [64].

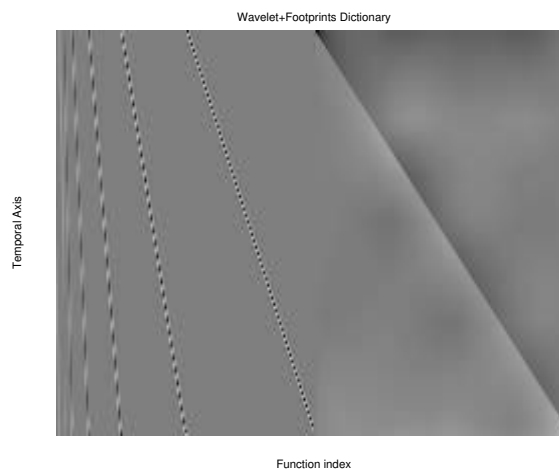


Figure 6.2: Dictionary formed by wavelets (left half) and its respective footprints for piecewise constant singularities (right half).

The estimation of the *a priori* information is based on a signal adaptive parametric model that establishes a relationship between the dictionary, its internal structure and the input data. Roughly speaking, the *a priori* model used here is composed of two steps: first, an estimate of the location of edges in the signal is generated; then, W is configured so that footprints are favored to describe discontinuities, while wavelets are privileged for smooth regions. For a more detailed explanation of the model configuration as well as for the parameter optimization, we refer to Section 6.4, Algorithm 6.1.

The signal f is decomposed by solving BPDN (problem (P_{2-1})) and WBPDN (P_{2-1}^w) , where the *a priori* knowledge is introduced. Both solutions are numerically found using Quadratic Programming techniques. The trade-off parameter γ controls the ℓ_1 norm of the coefficient vector and indirectly its sparseness. The signal approximations present many components with negligible values due to the numerical computation: a hard thresholding is thus performed in order to get rid of these insignificant elements. In this way, it is possible to measure the ℓ_0 norm of the vector \mathbf{b} . The data reported here refer to a threshold value of 10^{-9} . However, the question of how to fix such a threshold is in general still open. Of course, the reconstructions are computed starting from the thresholded coefficients. Fig. 6.3 shows the reconstructions of the input signal given by a 10-terms approximation found by BPDN and WBPDN. The left-hand side of Fig. 6.4 illustrates the mean square error of the approximations.

Let us call \mathbf{b}_* the approximation found by BPDN and \mathbf{b}_*^w the one found by WBPDN. As just explained, these vectors are thresholded removing the numerically negligible components, and in this way we are able to individuate a sparse support and thus a subset of the dictionary. Exactly as explained in Section 4.5.5 it is possible to use BPDN or WBPDN only for selecting a subset of the dictionary and then recompute the coefficients by a simple projection.

Let us label the sub-dictionary found by WBPDN with \mathcal{D}_*^w (composed by the atoms corresponding to the non-zero elements of \mathbf{b}_*^w). Once this is given, there are no guarantees that the coefficients that represent f are optimal (see Theorems 4.4 and 6.1). These are, thus, recomputed projecting the signal onto the subspace spanned by the atoms of \mathcal{D}_*^w and a new approximation of f named \mathbf{b}_{**}^w is found. Exactly the same is done for BPDN, ending up with a sub-dictionary \mathcal{D}_* and a new approximation \mathbf{b}_{**} . Of course, $\text{support}(\mathbf{b}_*) = \text{support}(\mathbf{b}_{**})$ and $\text{support}(\mathbf{b}_*^w) = \text{support}(\mathbf{b}_{**}^w)$. Formally the approximants found by BPDN and WBPDN after the projection step are respectively:

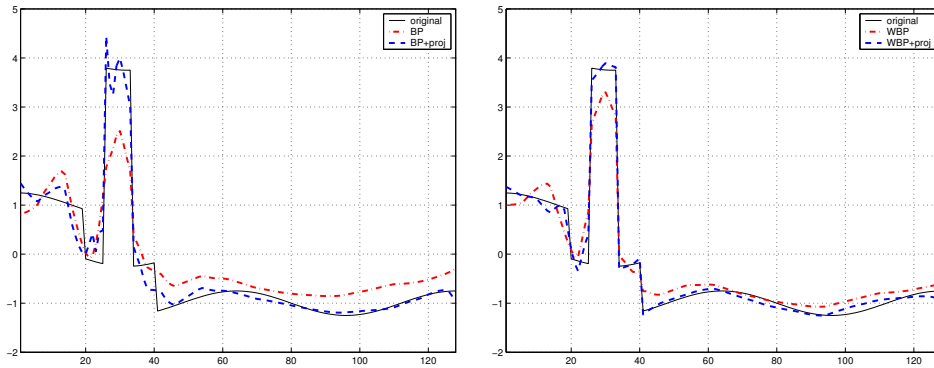


Figure 6.3: The original signal reconstructed from a 10-terms approximation computed by BPDN (left) and WBPDN (right). The comparison shows the improvement given by recomputing the projections once that the algorithm has selected a sub-dictionary.

$$\begin{aligned} f_{**} &= D_* D_*^+ f = D \mathbf{b}_{**} \text{ and} \\ f_{**}^w &= D_*^w (D_*^w)^+ f = D \mathbf{b}_{**}^w. \end{aligned} \quad (6.24)$$

Figures 6.3 and 6.4, show how this technique considerably improves the results obtained by solving problems (P_{2-1}) and (P_{2-1}^w) . Moreover they confirm the advantages of the weighted algorithm.

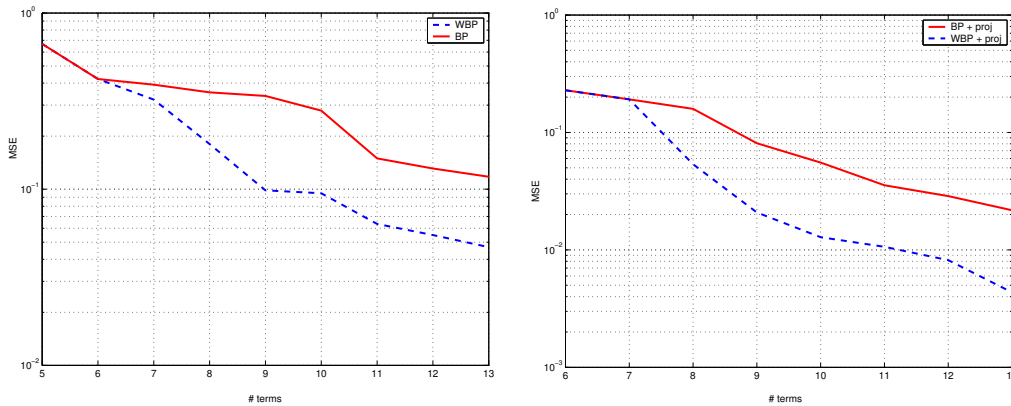


Figure 6.4: Errors (in log scale) of the m -term approximations found by BPDN and WBPDN. On the right-hand, the approximations are computed projecting the signal onto the sub-dictionary selected by the algorithm (see Eq. (6.24)).

6.3 Approximation by Weighted Greedy Algorithms

In this section we explore the effect of using *a priori* knowledge in greedy algorithms for the recovery of the best m -term approximant ($f_m^{opt} = D \cdot \mathbf{c}_{opt}$) of a signal f . This work appears in [49, 52], to which we refer for a more wide exposure of the theoretical results that we report here in short and for all the proofs.

Following [52], we define a Bayesian formulation of Matching Pursuit that we will call Weighted Matching Pursuit (Weighted-MP). In Sec. 4.4, we have recalled how MP and OMP use the scalar product as similarity measure for the selection of the most appropriate atom. This bears some

resemblance with searching the atom g_{i_k} with Maximum Likelihood for a given residual r_k . Indeed, the selection procedure in MP may be seen as a maximization of the probability $p(g_i|r_k)$, that is as considering $C(r_k, g_i) \sim p(g_i|r_k)$ in Eq. (4.19). At the same time, $|\langle r_k, g_i \rangle|$ may be intuitively considered as a measure of the conditional probability $p(r_k|g_i)$. In the case where all the atoms are *a priori* equiprobable, maximizing $p(r_k|g_i)$ is equivalent to maximize $p(g_i|r_k)$. Let us now study the case where the atoms do not necessarily have the same *a priori* probability to appear in the optimal set Γ , and let us assume that we have at our disposal a *prior* knowledge about the likelihood of each g_i . By means of the Bayes' Rule, when some *a priori* $p(g_i)$ is available, the probability to maximize becomes

$$p(g_i|r_k) = \frac{p(r_k|g_i)p(g_i)}{p(r_k)}, \quad (6.25)$$

where the denominator has a constant value once that r_k is given. Emulating this, the selection rule of MP can, thus, be modified multiplying the modulus of the scalar product by a weighting factor $w_i \in (0, 1]$, which depends on the atom index i . This is done in order to represent the insertion of some heuristic measure of prior information. Hence, now $C(r_k, g_i)$ in Eq. (4.19) can be considered such that:

$$C(r_k, g_i) = |\langle r_k, g_i \rangle| \cdot w_i.$$

We call this family of weighted greedy algorithms Weighted-MP. The Weighted-MP approach does not modify the projection step of the algorithm, allowing to freely select the MP or OMP projection strategy. For the sake of simplicity, Weighted-MP will be used in the remaining of the dissertation as a general term to refer to both projection approaches. The kind of projection will not be specified unless judged relevant. In this work, we assume that the *a priori* knowledge (appearing under the form of weights w_i) is independent of the iteration of the algorithm (hence, $p(r_k) = \text{constant}$). However, one could decide to update the atom weights at every iteration, leading to take also into account, in some way, $p(r_k)$. This would introduce more flexibility in the formulation of Weighted-MP.

Unit norm atoms are re-weighted according to some heuristic measure of prior information, which gives some hint about their likelihood to belong to the optimal set Γ . One may also interpret Weighted-MP as a greedy algorithm where the use of non-unit norm atoms within the dictionary is allowed. In the following, sufficient conditions for the recovery of a ‘‘correct’’ atom from the sparsest m -term approximant are established. Later, we study how *a priori* knowledge affects the rate of convergence of greedy algorithms, and finally, an example is presented. As we will see, Weighted-MP is able to perform better than Pure MP.

6.3.1 Influence of *a priori* Information on Recovery Conditions

We can now observe the behavior of greedy algorithms when *a priori* information taken into account.

Theorem 6.3 *Let $\{r_k\} : k \geq 0$, be the set of residuals generated by Weighted-MP in the approximation of a signal f , and let f_m^{opt} be its best m -term approximant. Then, for any positive integer m such that $\mu_1^w(m-1) + \mu_1^w(m) < 1 - \epsilon_{\max}$ and*

$$\|r_k\|_2^2 > \|f - f_m^{\text{opt}}\|_2^2 \left(1 + \frac{m(1 - (\mu_1^w(m-1) + \epsilon_{\max})) (w_{\Gamma}^{\max})^2}{(1 - (\mu_1^w(m-1) + \mu_1^w(m) + \epsilon_{\max}))^2} \right), \quad (6.26)$$

Weighted-MP will recover an atom that belongs to the optimal set Γ (in the sense of (6.1)).

In the case that f_m^{opt} can not be reached or just an approximate solution exists, a sub-optimality factor $\eta \geq 0$ can be introduced by substituting $\|f - f_m^{\text{opt}}\|_2^2$ by $\|f - f_m^{\text{opt}}\|_2^2 (1 + \eta)^2$ in Eq. (6.26).

Theorem 6.3 means that, if the approximation error at the k th iteration is still bigger than a certain quantity, then another term of the best m -term approximant can be recovered. This is similar to the result of [98], but here the use of *a priori* information results in a smaller bound. More terms may, thus, be recovered. Finally, the general effect of using *a priori* knowledge can be summarized by the following corollary.

Corollary 6.2 *Let $W(f, \mathcal{D})$ express a reliable a priori knowledge and assume $\alpha = 1$, then for any positive integer m such that $\mu_1(m-1) + \mu_1(m) \geq 1$ but $\mu_1^w(m-1) + \mu_1^w(m) < 1 - \epsilon_{max}$, Weighted-MP (unlike MP) will recover the atoms belonging to the best m -term approximant f_m^{opt} . Moreover, for any positive integer m such that $\mu_1^w(m-1) + \mu_1^w(m) + \epsilon_{max} \leq \mu_1(m-1) + \mu_1(m) < 1$, Weighted-MP has a weaker sufficient condition than MP for the recovery of correct atoms from the best m -term approximant. Hence, the correction factor of the right-hand side of expression (6.26) is equal or smaller in the weighted case for any value of $w_{\Gamma}^{max} \in (0, 1]$:*

$$\left(1 + \frac{m \left(1 - (\mu_1^w(m-1) + \epsilon_{max}) (w_{\Gamma}^{max})^2 \right)}{\left(1 - (\mu_1^w(m-1) + \mu_1^w(m) + \epsilon_{max}) \right)^2} \right) \leq \left(1 + \frac{m(1 - \mu_1(m-1))}{\left(1 - (\mu_1(m-1) + \mu_1(m)) \right)^2} \right). \quad (6.27)$$

Therefore, Weighted-MP is guaranteed to recover equally good or better approximants than classic MP when reliable *a priori* information is used (if $\mu_1^w(m) + \mu_1^w(m+1) + \epsilon_{max} < \mu_1(m-1) + \mu_1(m) < 1$, then the better behavior is guaranteed).

6.3.2 Rate of Convergence of Weighted-MP

The energy of the series of residuals r_k generated by the greedy algorithm progressively converges toward zero as k increases. In the same way, Weighted-MP with reliable *a priori* information is expected to have a better behavior and a faster convergence rate than the Weak-MP for the approximation case. A tighter measure of the dictionary coherence conditioned to the signal to be analyzed is available: $\mu_1^w(m)$ (where $\mu_1^w(m) \leq \mu_1(m)$). Then a better bound for the rate of convergence can be found for the case of Weighted-MP. To prove this, we follow the path suggested in [174] and [98], introducing as before the consideration of the *a priori* information in the formulation. The results formally show how Weighted-MP can outperform Weak-MP when the *a priori* knowledge is reliable.

Theorem 6.4 *Let $W(f, \mathcal{D})$ be a reliable a priori information matrix and $\{r_k\} : k \geq 0$ a sequence of residuals produced by Weighted-MP, then as long as $\|r_k\|_2^2$ satisfies Eq. (6.26), Weighted-MP picks up a correct atom and*

$$\left(\|r_k\|_2^2 - \|r_m^{opt}\|_2^2 \right) \leq \left(1 - \alpha^2 \frac{(1 - \mu_1^w(m-1) - \epsilon_{max})}{m} \right)^{k-l} \left(\|r_l\|_2^2 - \|r_m^{opt}\|_2^2 \right), \quad (6.28)$$

where $k \geq l$.

As observed for Theorem 6.3, if f_m^{opt} can not be reached or just an approximate solution exist, $\|r_m^{opt}\|_2^2$ is substituted by $\|r_m^{opt}\|_2^2 (1 + \eta)^2$ in Eq. (6.28).

Theorem 6.4 implies that the rate of convergence of Weighted-MP has an upper bound with exponential decay, as well as Weak-MP. Moreover, in the case where reliable *a priori* information is used, the bound appears to be lower. This result suggests that the convergence of suitably weighted greedy algorithms is faster than in the case of pure greedy algorithms. Of course, this is subject to the use of a model that puts in relation both the signal and dictionary.

Depending on the sufficient conditions previously specified, it will be possible to recover the optimal set Γ . However, it is not yet clear how long a non-orthogonalized greedy algorithm (Weighted-MP in our case) will last iterating over the optimal set of atoms in the approximation case. Let us define the number of correct iterations as follows:

Definition 6.5 Consider a Weighted-MP algorithm used for the approximation of signals. We define the number of provably correct steps N_m as the smallest positive integer such that

$$\|r_{N_m}\|_2^2 \leq \|f - f_m^{opt}\|_2^2 \left(1 + \frac{m \left(1 - (\mu_1^w(m-1) + \epsilon_{max}) (w_{\bar{\Gamma}}^{max})^2 \right)}{\left(1 - (\mu_1^w(m-1) + \mu_1^w(m) + \epsilon_{max}) \right)^2} \right),$$

which corresponds to the number of atoms belonging to the optimal set that can be recovered given a signal f , a dictionary \mathcal{D} and an a priori information matrix $W(f, \mathcal{D})$.

In the case of OMP and Weighted-OMP, N_m will be always smaller or equal to the cardinality of Γ . For Weak-MP and Weighted-MP, provided that $\mu_1^w(m-1) + \mu_1^w(m) + \epsilon_{max} < 1$, the provable number of correct iterations will depend on the final error of the best m -term approximation. In the following theorem, bounds on the quantity N_m are given for Weighted-MP.

Before stating the theorem, the reader should note that from now on, $w_{\Gamma_l}^{max}$ defines the same concept as in (6.3) for an optimal set of atoms Γ of size l , i.e. for Γ_l .

Theorem 6.5 Let $W(f, \mathcal{D})$ be a reliable a priori information and $\{r_k\} : k \geq 0$ a sequence of residuals produced by Weighted-MP when approximating f . Then, for any integer m such that $\mu_1^w(m-1) + \mu_1^w(m) + \epsilon_{max} < 1$, we have $N_1 \leq 1$ and for $m \geq 2$:

- if $3 \|r_1^{opt}\|_2^2 \geq m \cdot \|r_m^{opt}\|_2^2 (1 - \epsilon_{max_m}) \cdot (w_{\bar{\Gamma}}^{max})^2$, then

$$2 \leq N_m < 2 + \frac{m}{1 - \mu_1^w(m-1) - \epsilon_{max}} \log \left(\frac{3 \|r_1^{opt}\|_2^2}{m \cdot \|r_m^{opt}\|_2^2 (1 - \epsilon_{max_m}) \cdot (w_{\bar{\Gamma}}^{max})^2} \right). \quad (6.29)$$

- else $N_m \leq 1$.

From (6.29) we can draw that the upper bound on the provably correct number of steps N_m is tighter for Weighted-MP if a reliable a priori knowledge is used. Indeed, in accordance with Theorem 6.4, which states a tighter residual error convergence bound for Weighted-MP, one can also have a tighter estimate for Weighted-MP about which is the maximum number of good iterations the algorithm might do. If some a priori is available, some atom interactions will not influence $\mu_1^w(m-1)$ in Eq. (6.29), unlike in the case of Theorem 7 in [98] where $\mu_1(m-1)$ was used.

Moreover, in a situation where the reliable a priori model was discriminative enough, we are sure that there would be additional room for an improvement on the number of correct iterations recovered by the greedy algorithm with respect to [98]. The term $w_{\bar{\Gamma}}^{max}$ helps to increase the value of the bound, describing the fact that Weighted-MP can recover a higher number of correct iterations than MP. In addition, compared to the case when no a priori information is available, the condition for the validity of bound (6.29) is softened. The assumption of good discrimination capabilities of the a priori model is somehow unrealistic in practice, i.e. a small value for $w_{\bar{\Gamma}}^{max}$ indicates that the model can already discriminate between Γ and $\bar{\Gamma}$. Nevertheless, the result of Theorem 6.5, gives a better estimate on the upper bound of N_m thanks to the use of $\mu_1^w(m-1)$ instead of $\mu_1(m-1)$, and furthermore it suggests that using an a priori model should have a positive effect on the stability of

Weighted-MP. In practice, if the *prior* is capable to handle some punctual ambiguity that may affect the choice of the appropriate function at a given MP step, then the benefits for the convergence of the algorithm can be of extreme relevance. This can be the case even if the *a priori* model does not supply a good discrimination between Γ and $\bar{\Gamma}$. Examples in Sections 6.3.3 and 6.4 illustrate this situation.

6.3.3 Example: Use of Footprints and Weighted-OMP for Sparse Approximations

We examine again the example presented in Section 6.2.4, but this time using Weighted-MP. The dictionary and the input signal are illustrated in Section 6.2.4. For an explanation of the prior model and the extraction of the *a priori* matrix, see Sec. 6.4.

Figure 6.5 presents, from left to right, the original signal and the two approximants obtained by OMP without and with *a priori* information. The input signal has a polynomial degree which is higher than the number of vanishing moments of the *Symmlet-4*. With very few components, the algorithm benefits from the *a priori* information estimated from the signal, and gives a much better approximation. A more global view of this enhancement can be seen in Fig. 6.6 where the convergence of the approximation error is presented. The use of weights is definitively helpful and a considerable reduction of the error is achieved for a small number of terms.

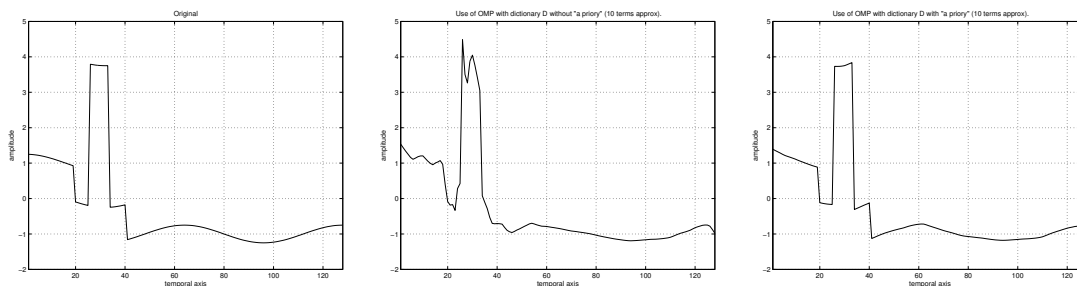


Figure 6.5: Comparison of OMP based approximation with 10 terms using the footprints dictionary (Fig. 6.2). Left: Original signal. Middle: “blind” OMP approximation. Right: OMP with prior knowledge of the footprints location.

6.4 Natural Signal Approximation with an *A Priori* Model

In this section we apply the methodology introduced in Sections 6.2 and 6.3 to natural signals. We also discuss the problem of finding reliable *a priori* information on a concrete example. Moreover, we show how the *a priori* weights can be automatically extracted from the data and optimized in order to maximize the performance of the weighted algorithms. We approximate several 1-D signals, extracted from a variety of columns of *cameraman* and *Lena* images that can be considered as piecewise-smooth, by using an overcomplete, coherent dictionary.

6.4.1 Modeling the Relation Signal-Dictionary

The dictionary is composed by the union of the *Symmlet-4* orthonormal basis, used to model smooth parts of the signal, and the set of piecewise-constant footprints meant to model discontinuities (see Sec. 6.2.4 and Fig. 6.2). Since the input signal has 256 samples, D is a matrix of size 256×512 . The

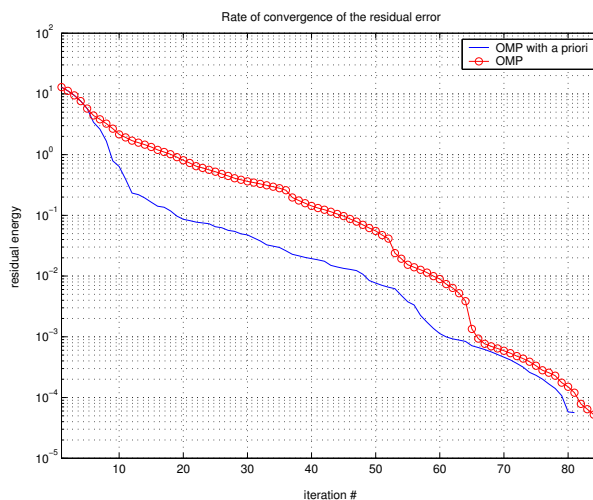


Figure 6.6: Rate of convergence of the error with respect to the iteration number in the experiment of Fig. 6.5

weighting matrix $W(f, D)$ is generated by means of an estimation of the locations where footprints are likely to be used, assuming that in such locations wavelets have less probability to appear. This discrimination does not penalize locations where a footprint is likely to be placed (thus the weighting factor remains 1). On the contrary, wavelets that overlap the footprint, as well as footprints considered unlikely to be used, get a penalizing factor $\beta \in (0, 1]$. The modeling of the interaction between the signal and the dictionary is performed using the Algorithm 6.1.

Algorithm 6.1: $W(f, D)$ estimation

Require: $\mathcal{D} = \mathcal{D}_{Symmlet} \cup \mathcal{D}_{Footprints}$, define a threshold λ , define a penalty factor β

- 1: $v_{diff} = D_{Footprints}^+ \cdot f$ {Footprints location estimation (edge detection)}
 - 2: Threshold v_{diff} by λ putting greater values to 1 or β otherwise.
 - 3: $W_{footprints}^{diag} = v_{diff}$ {Diagonal of the sub-matrix of $W(f, D)$ corresponding to footprints.}
 - 4: Create W_{wave}^{diag} s.t. all wavelets intersecting the found footprints locations equal β , set to 1 otherwise.
 - 5: $W(f, D) = \text{diag} \left(\begin{bmatrix} W_{wave}^{diag} & W_{footprints}^{diag} \end{bmatrix} \right)$;
-

As one can observe, two parameters configure the model that generates $W(f, D)$: a threshold λ and a penalty weight β . We will show later that these can be selected by an optimization procedure that minimizes the average energy of the approximation error.

6.4.2 Signal Approximation

We resume the general procedure for the signal approximation by these two steps:

1. Estimation of the *a priori* information from the “real world” signal using an *a priori* model.
2. Use of a weighted algorithm (greedy or relaxed) based on the estimated *a priori* knowledge to find the appropriate atoms subset. Optionally, once these have been selected, their coefficients can be computed again, by means of a simple projection.

Furthermore, an iterative version of this algorithm can be considered in order to optimize the parameters that configure the *a priori* model used in the first step (λ and β in our examples). This can be seen as a kind of Expectation Maximization algorithm. The simplest approach for parameter tuning can be a grid search, or a multi-scale grid search. Nevertheless, much more sophisticated and efficient search techniques may be used to optimize the *a priori* models. See [149] for some global optimization techniques.

6.4.3 Results

Now quantitative impact of using weighted algorithms is illustrated in terms of the residual error energy. Then, we describe how atoms can represent the main features of a signal, and finally, we explore the influence of tuning the two parameters that configure our penalty model.

Approximation Results with OMP

The improvement of Weighted-OMP in the case of sparse approximations is assessed by the rate of convergence of the residual energy, on the right-hand side of Fig. 6.7: the graph shows that after few iterations, Weighted-OMP selects better atoms than classic OMP. Hence the convergence of the error improves and this yields a gain of up to 2 dB.

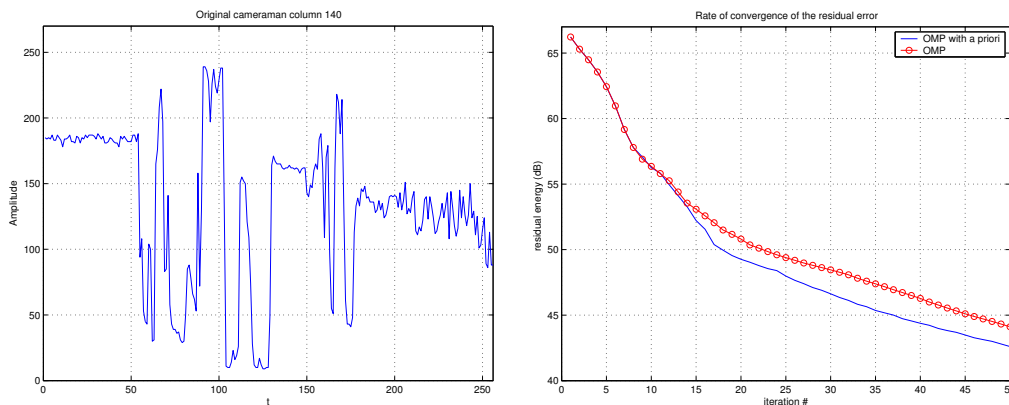


Figure 6.7: Experiment of approximating the 1-D signal extracted from the 140th column of *cameraman* (On the left). On the right, the rate of convergence of the residual error for OMP and Weighted-OMP.

We want to stress again that, extracting relevant footprints and wavelets by simply selecting those with higher *a priori* weights does not yield good sparse approximations. The *a priori* model is just supposed to give rough hints about which functions are useful for every particular signal feature. The *a priori* model is not supposed in any case to give a precise profile of the exact atoms to be used in a particular signal approximation. For instance, the weights computed in our example equal 1 for more than 200 functions, making thus impossible to use thresholding on W as a self-standing selection criterion. Indeed, the use of simple thresholding would imply that $\beta = 0$ in the model. As one can see in Fig. 6.10 (“probability weight” axis), $\beta = 0$ does not supply the best approximation error average. The model must be used in conjunction with the atom selection procedure of an appropriate nonlinear subset selection algorithm.

Approximation Results with BPDN

The same signal is now approximated by BPDN and WBPDN. As explained in Section 6.2.4, the pursuit algorithm is used only to select a dictionary subset and then the coefficients of the approximation are computed again, by means of a simple projection. Fig. 6.8 shows the decay of the error versus the number of atoms. It is clear how the use of the *a priori* helps the algorithm in finding a better approximation of the signal. The results concerning WBPDN are obtained by adopting a weighting matrix that corresponds to $\lambda = 90$ and $\beta = 0.2$. Notice that these values are not optimal for all the numbers of non-zero coefficients, as can be seen in the area between 34th and 43rd coefficients in the graph of Fig. 6.8. Better results can be achieved by tuning appropriately β and γ for any desired m .

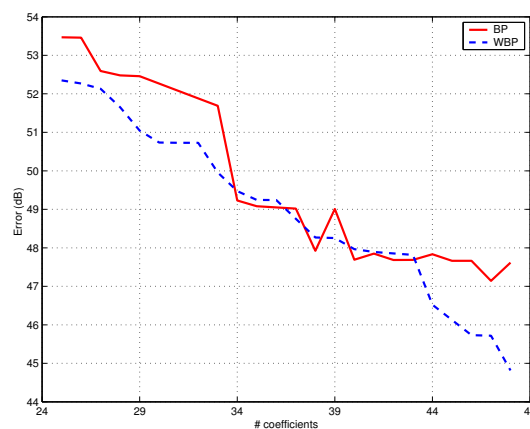


Figure 6.8: Error (in dB) obtained by BPDN and WBPDN. Both results are obtained by using quadratic programming for selecting a dictionary subset and then recomputing the coefficients by projecting the signal onto the span of the sub-dictionary. The procedure is illustrated in Sec. 6.2.4.

Capturing the Piecewise-smooth Component with Footprints Basis

Here, the results intend to underline the importance of selecting the appropriate atom to represent a particular signal feature. In the top row of Fig. 6.9 we can see the resulting approximants after 50 iterations of OMP (left) and Weighted-OMP (right). The result obtained by including the *a priori* is about 1.5 dB better than the one obtained by OMP. At this point, it is important to observe the bottom row of Fig. 6.9. These waveforms represent the signal components captured exclusively by the footprints and wavelet scaling functions. These components should correspond to the piecewise-smooth parts of the signal. However, in the case of OMP (bottom left) the piecewise-smooth component captured by footprints and low-pass functions is far from what one could expect. Intuitively one can understand that OMP is failing in the selection of atoms. On the other hand, the result obtained by Weighted-OMP (bottom right) clearly shows that footprints and *Symmlet-4* scaling functions capture a much more accurate approximant of the piecewise-smooth component of the signal. We can thus argue that a better approximation is achieved by using the *a priori* information, and this leads to a sparser approximation too.

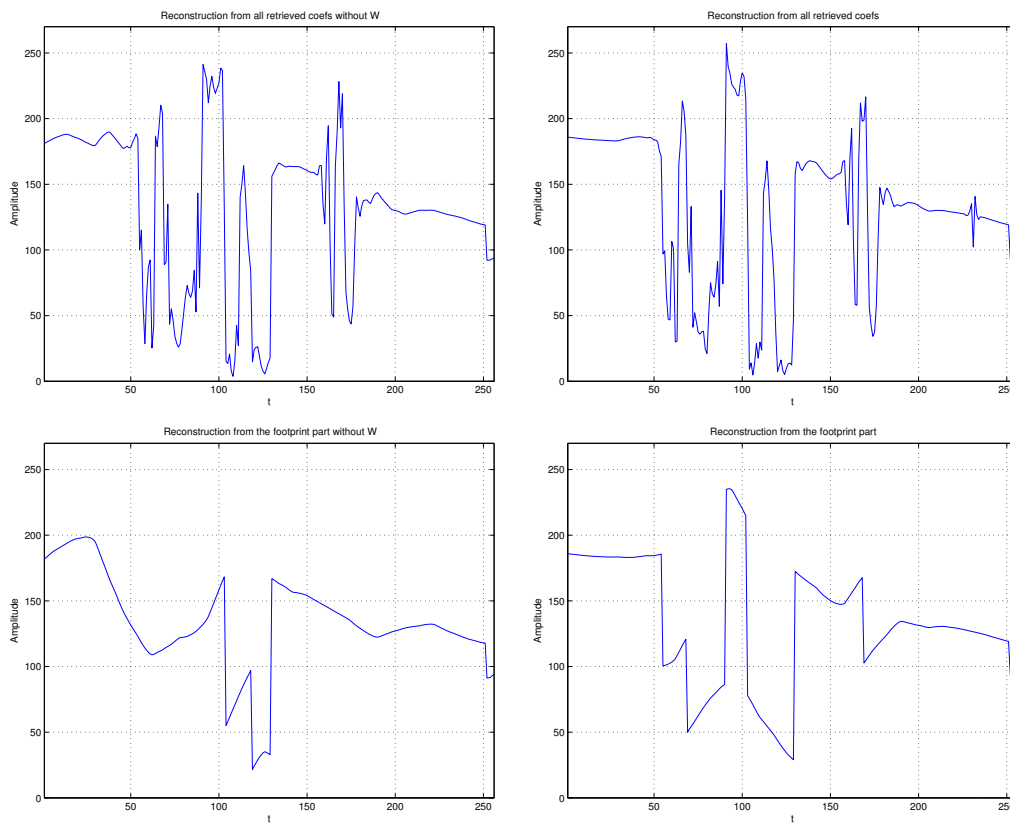


Figure 6.9: Top: Approximation after 50 iterations of OMP with (right) and without (left) *a priori* information. Bottom left: Signal components captured by *Symmlet* scaling functions and Footprints using OMP. Bottom right: Signal components captured by *Symmlet* scaling functions and Footprints using Weighted-OMP.

Parameter Search

Let us now consider the influence of the parameters λ and β in the average quadratic error of the residues obtained by Weighted-OMP, i.e.

$$E \{r_k | \lambda', \beta'\} = \frac{\sum_{k=0}^{K-1} \|r_k\|_2^2}{K}, \quad (6.30)$$

such that r_k is obtained fixing $\lambda = \lambda'$ and $\beta = \beta'$.

In Fig. 6.10, the magnitude of Eq. (6.30) is shown as a function of λ (model threshold) and β (probability weight). The lower the value of $E \{r_k | \lambda', \beta'\}$, the higher the probability of the parameters to be the good ones.

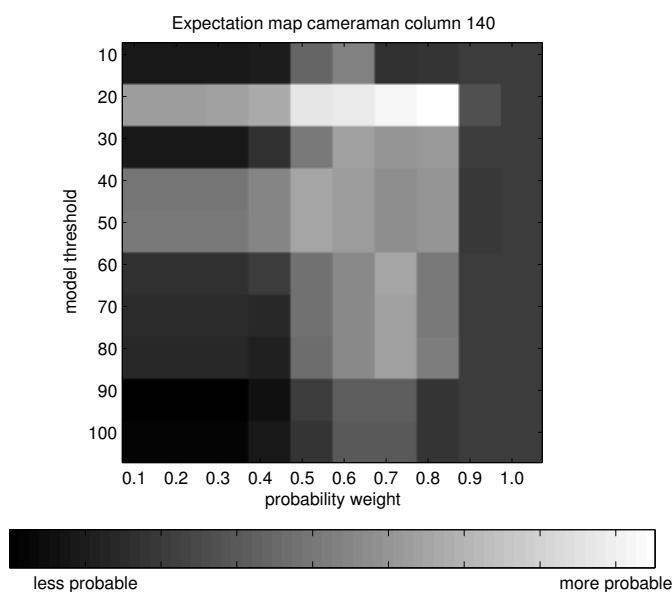


Figure 6.10: Representation of the expectation map depending on the parameters that configure the *a priori* model in the experiment set up in Fig. 6.7. The expectation corresponds to the average energy of the residual error.

Hence, it can be easily observed that a unique global optimum in the parameter space exists. In this example, thanks to the convexity of the solution, the set of parameters which best fit the data model can be found by some iterative procedure. However the choice of the optimal values for λ and β is not straightforward since it is signal-dependent. Additional experimental results may be found in [50, 52].

6.5 Including *a Priori* Information in Exact Representation Problems

Up to now, in this chapter we have analyzed the effects of introducing *a priori* information in sparse approximation algorithms. This section focuses on the exact representation problem. To show the impact of reliable *priors*, we study the behavior of Weighted Basis Pursuit and Weighted-MP.

6.5.1 Exact Recovery via WBP

The BP principle selects the signal representation \mathbf{b} that has minimal ℓ_1 norm, as shown in Eq. (4.31). A variation of this method which allows to take into account the likelihood matrix $W(f, \mathcal{D})$, is given by Weighted Basis Pursuit (WBP). This method consists on minimizing the ℓ_1 norm of a weighted vector, leaving the constraints unchanged:

$$(P_1^w) \quad \arg \min_{\mathbf{b} \in \mathbb{R}^d} \|W^{-1}\mathbf{b}\|_1 \quad \text{s.t.} \quad \mathcal{D}\mathbf{b} = f. \quad (6.31)$$

We recall that the entries of $W(f, \mathcal{D})$ are in $(0, 1]$. In this way the atoms with low probability to be selected are penalized by inducing a small weighting factor in W . It can be proved that WBP can be equivalently reformulated as a Linear Programming problem [85], just as BP.

In [117] the authors propose a minimization problem formally identical to (6.31), however it is not aimed at improving the ℓ_0 sparsity of the representation but at using a different cost function, for example considering the Total Variation. Moreover, they also study the case where the matrix that plays the role of W^{-1} is not rectangular.

It is possible to establish an exact recovery condition for Weighted Basis Pursuit. Next theorem basically states which is the sufficient condition such that, given the weights $W(f, \mathcal{D})$, WBP is a correct algorithm for recovering an exact sparse superposition of m atoms from \mathcal{D} . Let us just point out that, in the following, we will call \mathbf{b}_Γ the vector giving the optimal signal representation. It thus contains the coefficients corresponding to the functions in D_Γ and its size is m .

Theorem 6.6 *Given a dictionary \mathcal{D} and an a priori matrix $W(f, \mathcal{D})$, Weighted Basis Pursuit recovers the optimal representation of a sparse signal $f = D_\Gamma \mathbf{b}_\Gamma$ if:*

$$\sup_{g_i \in D_\Gamma} \left\| (D_\Gamma W_\Gamma)^+ g_i \cdot w_i \right\|_1 < 1. \quad (6.32)$$

Proof: Suppose that the optimal representation of f is given by $D_\Gamma \mathbf{b}_\Gamma$ and that condition (6.32) is respected. Suppose also that there exists a different representation $f = D_{alt} \mathbf{b}_{alt}$: there should be at least one atom that belongs to D_{alt} but does not appear in D_Γ . Let us call it g_x . What we want to prove is that

$$\|W_\Gamma^{-1} \mathbf{b}_\Gamma\|_1 < \|W_{alt}^{-1} \mathbf{b}_{alt}\|_1, \quad (6.33)$$

where W_{alt} is the square diagonal matrix containing the weights corresponding to the atoms in D_{alt} .

$$\begin{aligned} \|W_\Gamma^{-1} \mathbf{b}_\Gamma\|_1 &= \|W_\Gamma^{-1} D_\Gamma^+ D_\Gamma \mathbf{b}_\Gamma\|_1 = \\ & \|W_\Gamma^{-1} D_\Gamma^+ D_{alt} \mathbf{b}_{alt}\|_1 = \\ & \|W_\Gamma^{-1} D_\Gamma^+ D_{alt} W_{alt} W_{alt}^{-1} \mathbf{b}_{alt}\|_1 = \\ & \left\| (D_\Gamma W_\Gamma)^+ (D_{alt} W_{alt}) W_{alt}^{-1} \mathbf{b}_{alt} \right\|_1. \end{aligned}$$

If the columns of $M = (D_\Gamma W_\Gamma)^+ (D_{alt} W_{alt})$ do not have identical ℓ_1 norms, using Lemma 3.4 in [174] we can state that:

$$\|W_\Gamma^{-1} \mathbf{b}_{opt}\|_1 < \|M\|_{1,1} \cdot \|W_{alt}^{-1} \mathbf{b}_{alt}\|_1,$$

but

$$\|M\|_{1,1} = \sup_{g_i \in D_{alt}} \left\| (D_\Gamma W_\Gamma)^+ g_i \cdot w_i \right\|_1.$$

There are now two possibilities: either $g_i \in D_\Gamma$ and so the supremum is ≤ 1 , either $g_i \in D_{\bar{\Gamma}}$ and so the supremum is smaller than 1 thanks to (6.32). In both cases we obtain that (6.33) is respected.

On the other hand, if all the columns of the matrix M have the same ℓ_1 norm, this must equal $\left\| (D_\Gamma W_\Gamma)^+ g_x \cdot w_x \right\|_1$, where w_x is the weight corresponding to g_x . Hypothesis (6.32) ensures that this norm is strictly smaller than 1, thus:

$$\|W_\Gamma^{-1} \mathbf{b}_\Gamma\|_1 \leq \|M\|_{1,1} \cdot \|W_{alt}^{-1} \mathbf{b}_{alt}\|_1,$$

but this time $\|M\|_{1,1} < 1$. We can therefore conclude that in both cases (6.33) is valid and so WBP finds the sparsest solution. ■

6.5.2 Exact Recovery via Weighted-MP

It is possible to prove that Weighted-MP/OMP is also able to give an exact representation of a signal, i.e. to find all the “good” atoms. In this section we report the main theoretical result that describes the capacity of Weighted-MP/OMP to exactly recover a given signal. It establishes the Exact Recovery Condition for Weighted-MP/OMP, a sufficient condition for recovering at each iteration the atoms in the optimal index subset Γ . However, we do not include the proof, which can be found in [51] and [49], together with a more detailed explanation.

Theorem 6.7 *Given an a priori matrix $W(f, \mathcal{D})$ and a sub-optimality search factor $\alpha \in (0, 1]$, then, for any index set Γ such that $f \in \text{span}(g_\gamma, \gamma \in \Gamma)$, Weighted-MP/OMP will recover a “correct” atom at each iteration if*

$$\sup_{g_i \in D_{\bar{\Gamma}}} \left\| (D_\Gamma W_\Gamma)^+ g_i \cdot w_i \right\|_1 < \alpha. \quad (6.34)$$

Theorem 6.7 states, as depicted by (6.34), that the use of *a priori* weights will help meeting the sufficient condition that guarantees that a greedy algorithm will recover the elements of the sparsest representation of f . Indeed, as can be observed in (6.34), given a dictionary and an appropriate W_Γ associated to f , the weights that multiply each $g_i \in D_{\bar{\Gamma}}$ may help reducing the supremum in Eq. (6.34).

Finally, we can observe that there is a single sufficient condition, valid for both WBP and Weighted-MP/OMP, for recovering the “correct” set of atoms involved in the optimal representation of a signal. Such condition corresponds to Equation (4.33) for the case without weights.

6.5.3 Exact Recovery Bounds for WBP and Weighted Greedy Algorithms

In the following we provide a sufficient condition based on the weighted cumulative coherence for the recovery of the sparsest exact representation. Such condition is valid for both WBP and Weighted Greedy Algorithms.

Theorem 6.8 *Let $W(f, \mathcal{D})$ be the data dependent weighting matrix and let $\epsilon_{max} = \sup_{\gamma \in \Gamma} |1 - w_\gamma^2|$.*

If, for any index set Γ of size at most m , such that $f = \sum_{\gamma \in \Gamma} b_\gamma g_\gamma$, we have

$$\mu_1^w(m) + \mu_1^w(m-1) < 1 - \epsilon_{max}, \quad (6.35)$$

then (6.32) holds and WBP recovers the optimal representation of the sparse signal f . Furthermore, if

$$\frac{\mu_1^w(m)}{1 - (\mu_1^w(m-1) + \epsilon_{max})} < \alpha \quad (6.36)$$

is also enforced, then (6.34) holds and Weighted-MP will pick up an atom belonging to the optimal set Γ at each step. Moreover, Weighted-OMP will exactly recover the sparsest representation of f .

Since $\mu_1^w(m) \leq \mu_1(m)$, one can intuitively see that a reliable *a priori* knowledge can help a greedy algorithm or BP when the dictionary does not satisfy the hypothesis of Eq. (4.33). This will be possible when the weights corresponding to the atoms in $D \setminus D_\Gamma$ are sufficiently small.

Proof: Theorems 6.7 and 6.6 give the conditions under which Weighted *Weak*-MP and WBP recover the optimal set of atoms. In this proof the factor α is conserved independently of the algorithm in use. Note that for the particular results of WBP and Weighted-MP/OMP this value equals 1.

Starting from (6.34) and following the procedure suggested in [174] an upper bound based on μ_1^w can be obtained:

$$\begin{aligned} & \sup_{g_i \in D_{\bar{\Gamma}}} \left\| (D_\Gamma W_\Gamma)^+ g_i \cdot w_i \right\|_1 = \\ & \sup_{g_i \in D_{\bar{\Gamma}}} \left\| \left((D_\Gamma W_\Gamma^T)^T (D_\Gamma W_\Gamma^T) \right)^{-1} (W_\Gamma D_\Gamma^T) g_i \cdot w_i \right\|_1 \leq \\ & \left\| \left((W_\Gamma D_\Gamma^T) (W_\Gamma D_\Gamma^T)^T \right)^{-1} \right\|_{1,1} \cdot \sup_{g_i \in D_{\bar{\Gamma}}} \left\| (W_\Gamma D_\Gamma^T) g_i \cdot w_i \right\|_1. \end{aligned} \quad (6.37)$$

The first term on the right hand side of the inequality corresponds to the 1, 1-norm of the inverse Gram matrix of the weighted sub-dictionary of optimal functions. This can be expressed as:

$$\left((W_\Gamma D_\Gamma^T) (W_\Gamma D_\Gamma^T)^T \right)^{-1} = (I + A_w)^{-1}, \quad (6.38)$$

where I denotes the identity matrix and A_w is a symmetric matrix. Due to the diagonal weight matrices W_Γ , the matrix A_w is not composed only of the off-diagonal elements. Adding and subtracting the identity matrix, we can rewrite (6.38) in the following way:

$$(I + A_w)^{-1} = \left(I + \left((W_\Gamma D_\Gamma^T) (W_\Gamma D_\Gamma^T)^T - I \right) \right)^{-1}.$$

Akin to [174] this can be expanded by means of *Neumann* series [99] and, if $\|A_w\|_{1,1} < 1$, we have:

$$\begin{aligned} \left\| (I + A_w)^{-1} \right\|_{1,1} &= \left\| \sum_{k=0}^{\infty} (-A_w)^k \right\|_{1,1} \\ &\leq \sum_{k=0}^{\infty} \|A_w\|_{1,1}^k = \frac{1}{1 - \|A_w\|_{1,1}}. \end{aligned}$$

Thus,

$$\left\| \left((W_\Gamma D_\Gamma^T) (W_\Gamma D_\Gamma^T)^T \right)^{-1} \right\|_{1,1} \leq \frac{1}{1 - \|A_w\|_{1,1}}. \quad (6.39)$$

The 1, 1-norm of A_w can be expressed as:

$$\|A_w\|_{1,1} = \sup_{g_\gamma \in D_\Gamma} \left[\sum_{l \neq \gamma} | \langle g_l, g_\gamma \rangle | \cdot w_l \cdot w_\gamma + |1 - w_\gamma^2| \right], \quad (6.40)$$

where the summation comes from the off-diagonal elements and the last term comes from the diagonal part. Note that for convergence of the *Neumann* series we need $\|A_w\|_{1,1} < 1$. This is ensured by hypothesis since $\|A_w\|_{1,1} \leq \mu_1^w(m-1) + \epsilon_{max}$ and

$$\mu_1^w(m-1) + \epsilon_{max} < 1$$

by (6.35) and (6.36). From (6.39) it follows that:

$$\left\| \left((W_\Gamma D_\Gamma^T) (W_\Gamma D_\Gamma^T)^T \right)^{-1} \right\|_{1,1} \leq \frac{1}{1 - (\mu_1^w(m-1) + \epsilon_{max})}. \quad (6.41)$$

Coming back to Eq. (6.37), the second term can be bounded as

$$\sup_{g_i \in D_\Gamma} \left\| (W_\Gamma D_\Gamma^T) g_i \cdot w_i \right\|_1 \leq \mu_1^w(m). \quad (6.42)$$

Finally, from (6.41) and (6.42) we obtain

$$\frac{\mu_1^w(m)}{1 - (\mu_1^w(m-1) + \epsilon_{max})} < \alpha, \quad (6.43)$$

and this proves the theorem. ■

Since $\mu_1^w(m) \leq \mu_1(m)$, we claim that considering reliable *a priori* information can help a dictionary unable to satisfy Eq. (4.33) to recover the right set of functions. In other words, reliable weights allow for using less incoherent dictionaries.

Corollary 6.3 *Given a dictionary \mathcal{D} and the data dependent diagonal matrix $W(f, \mathcal{D})$, where $w_i \in (0, 1]$, we can state the following:*

- *For a Weighted MP/OMP with weakness $\alpha = 1$ and WBP a better behavior in the recovery of exact sparse representations is expected with respect to the classical algorithms if:*

$$\begin{aligned} \mu_1^w(m) + \mu_1^w(m-1) &< 1 - \epsilon_{max} \\ \text{and} \\ \mu_1(m) + \mu_1(m-1) &\geq 1. \end{aligned}$$

- *For a Weighted Weak-MP a better behavior in the recovery of exact sparse representations is expected with respect to the classical algorithms if:*

$$\begin{aligned} \frac{\mu_1^w(m)}{1 - (\mu_1^w(m-1) + \epsilon_{max})} &< \alpha \\ \text{and} \\ \frac{\mu_1(m)}{1 - \mu_1(m-1)} &\geq \alpha. \end{aligned}$$

Note that, when no *a priori* information is available (i.e. $W(f, \mathcal{D}) = I$), and consequently $\epsilon_{max} = 0$, Theorem 6.8 boils down to the results found by Tropp [174] and stated in Chapter 4.

6.5.4 A Toy Example for WBP in \mathbb{R}^5

Let us now illustrate Theorems 6.6 and 6.8 with a toy example of exact signal recovery via (Weighted) Basis Pursuit. Suppose we have a signal $f = [0, M, A, M, 0]' \in \mathbb{R}^5$ depicted in Figure 6.11 and we want to decompose it with BP over the following dictionary $\mathcal{D} = \{g_i\}_{i=1,\dots,10}$:

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 1 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 \end{pmatrix}.$$

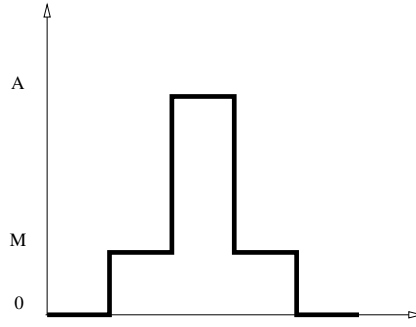


Figure 6.11: Signal $f \in \mathbb{R}^5$ to decompose over \mathcal{D}

The signal f has, of course, multiple representations over \mathcal{D} ; let us focus on two of them, setting $M = 1$ and $A = 3$:

$$\begin{aligned} f &= (g_3, g_{10}) \cdot \begin{pmatrix} 3 \\ \sqrt{2} \end{pmatrix} = D_{\Gamma} \cdot \mathbf{b}_{\Gamma} \\ &= (g_3, g_7, g_8) \cdot \begin{pmatrix} 1 \\ \sqrt{2} \\ \sqrt{2} \end{pmatrix} = D_{alt} \cdot \mathbf{b}_{alt}. \end{aligned} \tag{6.44}$$

Computing (4.33) for $m = 2$ we obtain a value around 2.1, thus bigger than 1. Hence, we have no guarantee that BP finds the sparsest solution, and in fact, BP selects the second representation in (6.44) which has a smaller ℓ_1 norm.

Let us now insert a weighting matrix W with the diagonal elements w_i equal to:

$$w_i = \begin{cases} 1 & \text{if } i = 1, 10 \\ 0.95 & \text{if } i = 3 \\ v < 1 & \text{otherwise} \end{cases}. \tag{6.45}$$

We know that the optimal support is given by $\Gamma = \{3, 10\}$. In this situation we can compute ϵ_{max} using Eq. (6.4), obtaining $\epsilon_{max} = 1 - w_3^2 = 0.0975$. Let us now set $v = 0.4$: we thus have $\mu_1^w(2) + \mu_1^w(1) \simeq 0.85 < 1 - \epsilon_{max}$. Therefore the hypothesis of Theorem 6.8 is respected and in fact WBP selects the sparsest representation.

This toy example shows how the use of *a priori* knowledge can lead the decomposition algorithm to find the sparsest solution. Here we do not question how we obtained the *a priori* provided by W , but note that the weights themselves do not contain sufficient information to select the optimal

subset of atoms. As a matter of fact, just selection the waveform corresponding to the biggest weights would have not provided the set $\{g_3, g_{10}\}$, but the set $\{g_1, g_{10}\}$.

We conclude by observing that WBP is able to find the sparsest signal representation of f even with a higher value of v , for example $v = 0.8$. However in this case $\mu_1^w(2) + \mu_1^w(1) \simeq 1.48$ and the hypothesis of Theorem 6.8 are no longer verified. This bears out the fact that the condition of Theorem 6.8 is quite pessimistic. The same is valid for the the case without weights in Eq. (4.33).

6.6 Applications

In this section we have a closer look to how the framework of weighted algorithm can be applied to solve real problems. We work with two very different kinds of structured signals: first images, then Electrocardiograms (ECG). These examples also give practical ways of computing the weights exploiting the *a priori* information we have about the signals.

6.6.1 Images

Following the WBPDN paradigm, we can decompose images over a redundant multi-component dictionary composed by a sub-dictionary \mathcal{D}_{edge} aimed at representing the edges and one \mathcal{D}_{smooth} for representing the smooth parts.

BPDN and its weighted version can be solved by QP, but if the size of the signal is big and the dictionary highly redundant the complexity is indeed prohibitive. This makes practically infeasible working with images and dictionaries similar to the ones introduced in Section 3.2. Solutions to this problem can be either splitting the image into blocks or partitioning the dictionary and minimizing the cost function in several step. The first method has the drawback of introducing blocking artifacts, moreover the possibility to catch long structures (longer than the block size) runs out. The second method is similar to the one used in the previously cited papers [160, 165, 166] and suffers from some restrictions concerning the convergence, as explained in Section 4.

Given the too high complexity of the problem, in this section we just give an example with a small toy image of size 16×16 . Furthermore, we use a much smaller dictionary. More in details, \mathcal{D}_{edge} is generated by translating, anisotropically scaling and rotating a 2-D mother function on the model of the dictionary presented in Section 3.2. Only one generating function is used (the one in Eq. (3.44)), the number of rotations and scaling factor is strongly reduced and the bending avoided. The sub-dictionary \mathcal{D}_{smooth} is composed by translated and isotropically scaled versions of a bidimensional Gaussian (see Eq. (3.51)).

Edginess

The algorithm we adopt for computing the weights is based on the *edginess* of the image, i.e. a quantity that says how much a certain position in the image can be considered like an edge. From our point of view we can interpret a (non binary) edginess value as an indication about how much a given location (x_1, x_2) is likely to be a center of an edge-oriented basis function.

The edginess is computed through the analysis of the dual local autocovariance matrix, following the method of [38, 182]. Differences in the eigenvalues of an autocovariance matrix indicate directions at which the local Fourier power spectrum of a function is slowly decreasing. It is therefore possible to assign a certain degree of edginess to any location in an image by looking at the relationship between these eigenvalues. Thanks to a slight variation of the method described in [38] we define an edginess $e(x_1, x_2)$ with values in $[0, 1]$ which tends to 1 when the point (x_1, x_2) has high probability

to lie on an edge:

$$e(x_1, x_2) = 1 - \left(\frac{\lambda_{min}}{\lambda_{max}} \right)^2. \quad (6.46)$$

λ_{min} and λ_{max} are respectively the minimum and maximum eigenvalue of the dual local autocovariance matrix centered in (x_1, x_2) .

Since we are working with a dictionary composed of functions well localized in the space, we can assign $e(x_1, x_2)$ as weight to the functions of \mathcal{D}_{edge} centered in (x_1, x_2) . Consequently the corresponding functions in \mathcal{D}_{smooth} will be weighted by $1 - e(x_1, x_2)$. In practice, as done in Algorithm 6.1 the edginess value above a certain value are set to 1. Note that according to the definition of Weighting matrix, its diagonal elements cannot be zero, because this will cause a problem in Eqs. (6.6) and (6.31). Practically setting w_i to zero is equivalent to removing the atom g_i from the dictionary.

Results

We now briefly show some results, approximating the 16×16 image on the left-hand side of Figure 6.12 with 31 atoms from the MCD. The right-hand side of the figure displays the edginess computed by using Eq. (6.46).

Figure 6.13 shows the image reconstructed by the coefficient selected by WBPDN and BPDN. The MSE is respectively 42.7 and 89.5, reflecting the advantages of properly exploiting the *a priori* information coming from the edginess. The gain of WBPDN is in fact given by the placement of more atoms on the contours. Remark that the recovery conditions are not satisfied for BPDN nor for WBPDN, so there is no guarantee that they find the optimal approximation. Incidentally, we observe that WBPDN is also faster than its non weighted version. Such speed-up can be explained by the fact that the convergence to the final solution is helped by the weights. Anyway this depends on the implementation strategy adopted for solving the QP problem and it is not necessarily always true.

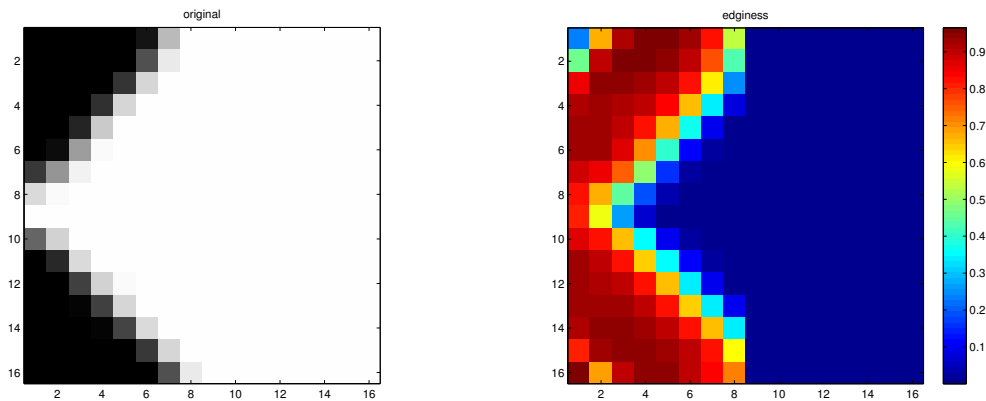


Figure 6.12: Original image and its edginess.

Of course there exist many other methods to measure the edginess and nothing constraint us to this particular technique and for example we can make use of the method of Canny [30]. However, the technique of Czaja and Wickerhauser that we adopted here can offer some interesting development. In fact, the dual local autocovariance matrix which determines the edginess through Eq. (6.46) can

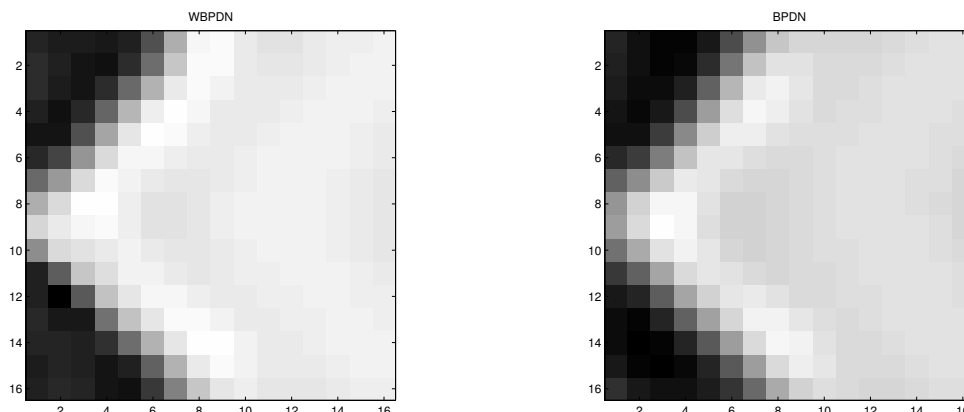


Figure 6.13: Images approximated by 31 atoms selected by WBPDN (left) and BPDN (right).

also give informations about the orientation of the edge. Again in [38] it is proved that the eigenvector of the larger eigenvalue will be normal to the edge (of course when such a normal exists). Thus, one can determine the edginess of each pixel and at the same time estimate the direction of the edge. We can then imagine a more general way to assign weights to the atoms: if we have a sub-dictionary parametrized as follows

$$\mathcal{D}_j = \{g_{(\vec{b}, \theta, \vec{a}, \dots)}(x_1, x_2)\}_{\vec{b}, \theta, \vec{a}, \dots}, \quad (6.47)$$

we can also consider the weights as:

$$W = W(\vec{b}, \theta, \vec{a}, \dots). \quad (6.48)$$

In this case the weights do not only depend on the location of the atoms but potentially on all its parameters. This model can certainly expand the horizon of the framework and improve the power of the weighted algorithms.

6.6.2 Electrocardiograms

The standard 12-lead ECG is a representation of the electrical activity of the heart, recorded from electrodes on the body surface. In Figure 6.14 we can see the evolution of the electrical activity during a healthy cardiac activation sequence (sinus rhythm) by means of its ECG representation. The depolarization of the atria manifests itself as the P wave and the depolarization of the ventricles causes the feature known as the QRS complex. The subsequent repolarization of the ventricular mass produces the T wave and the cardiac cycle concludes, while the repolarization of the atrium is hidden in the QRS complex. The U wave in the figure rises from the late ventricular repolarization which, however, is not always present and can be neglected for our purposes.

The atrial fibrillation (AF) is a supraventricular arrhythmia associated with the asynchronous contraction of the atrial muscle fibers. Without giving further clinical details which can be found for example in [186] we just observe that on the ECG, AF is described by the replacement of consistent P waves by rapid oscillations or fibrillatory waves that vary in size, shape, and timing, associated with an irregular, frequently rapid ventricular response.

The suitable analysis and characterization of atrial fibrillation from ECG recordings needs a previous isolation of the atrial activity component, due to the much higher amplitude of the electrical ventricular activity. Unfortunately, this low amplitude and the fact that both signals possess

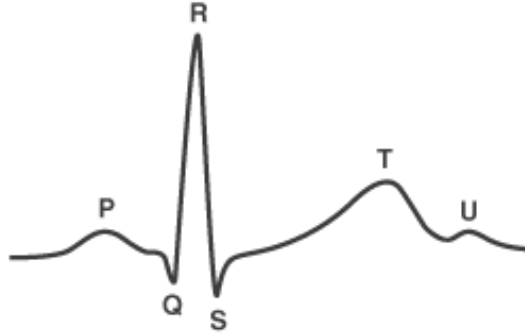


Figure 6.14: A scheme of an ECG signal exhibiting normal sinus rhythm.

spectral distributions that notably overlap, making linear filtering solutions unsuccessful, hinder this operation. Several methods have already been proposed to isolate the atrial activity when atrial fibrillation occurs. For example we can cite [31, 111, 112, 167].

In the following, we show how this problem can be treated as a source separation issue. Starting from the ECG signal, we aim at obtaining two different signals containing the isolated components of the ventricular activity (VA) and the atrial activity (AA). This task turns out to be very challenging since both components overlap in the frequency domain and are not orthogonal.

We model the input ECG signal f as a superposition of signal containing only AA and a signal containing only VA:

$$f \simeq f_{AA} + f_{VA}. \quad (6.49)$$

This model is the counterpart of Eq. (3.48) for images and it thus opens for the use of a multi-component dictionary. Suppose we have a MCD composed by two sub-dictionaries: \mathcal{D}_{VA} suited for representing the ventricular activity and \mathcal{D}_{AA} , of course, suited for representing the atrial activity.

We can therefore use an analysis algorithm to obtain a sparse m -term approximation (\mathbf{b}) of an ECG signal:

$$f \simeq D \cdot \mathbf{b} = D_{AA} \cdot \mathbf{b}_{AA} + D_{VA} \cdot \mathbf{b}_{VA}.$$

Observe that \mathbf{b} is composed of two parts, containing the coefficients concerning \mathcal{D}_{AA} and \mathcal{D}_{VA} . Now we can just recover the two components of the ECG signal by means of a simple reconstruction using only one sub-dictionary. Formally:

$$\begin{aligned} f_{AA} &\simeq D_{AA} \cdot \mathbf{b}_{AA} \\ f_{VA} &\simeq D_{VA} \cdot \mathbf{b}_{VA}. \end{aligned} \quad (6.50)$$

Dictionary design

The sub-dictionary oriented to represent the ventricular activity is generated by all possible translation of Generalized Gaussian functions:

$$g_{VA}(t) = C_1 \exp \left(- \left(\frac{|t-p|}{\alpha} \right)^\beta \right), \quad (6.51)$$

where C_1 is a normalizing constant, α determines the scale and β the peakiness. This waveform allows to well approximate the structure of a VA complex using few atoms. Fig. 6.15 shows on the left a QRST complex and its approximation by using only 3 atoms. With respect to the Gabor

function, the parameter β lets us approximate with more accuracy the Q and R peaks. The values of the two parameters of the generalized Gaussian have been chosen heuristically after an extensive set of tests [125]: $\alpha \in \{3, 4, 5, 6, 7\}$ for the Q and R waves and $\alpha \in \{49, 50, 51, 52, 53, 54\}$ for the T wave, while $\beta \in \{1.5, 1.6, \dots, 2.2\}$. Together with p , this makes \mathcal{D}_{VA} highly coherent, but also very flexible for VA approximation. However, such dictionary is far from being optimal, and several improvements are still possible, mainly concerning the approximation of T waves.

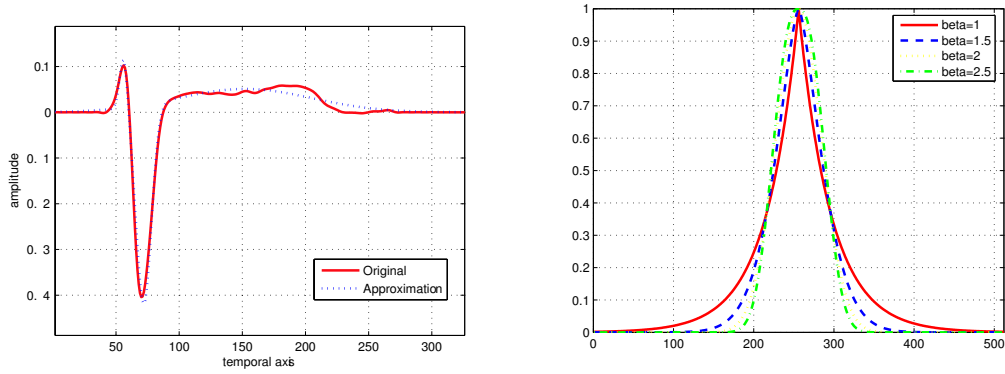


Figure 6.15: Left: QRST VA complex and its approximation using 3 atoms. Right: Effect of β on the GGF (see Eq. (6.51)).

The sub-dictionary designed to catch the structures of the AA is built by scaling, translating and modulating a Gabor function, its atoms having the following form:

$$g_{AA}(t) = C_2 \exp\left(-\left(\frac{t-p}{\alpha\sqrt{2}}\right)^2\right) \cos\left(\frac{2\pi k(t-p)}{N} - \Delta\psi\right), \quad (6.52)$$

where C_2 is a normalizing constant, n the length of the signal, α tunes the scale, k the frequency and $\Delta\psi$ the phase. The values of these parameters has been determined through a technical analysis and experiments on several real and simulated ECG signals. This waveform is specially adapted for AA approximation. Indeed, as can be observed in Fig. 6.16, fibrillating AA is of oscillatory nature, which is a perfect fit for the optimal spatio-temporal frequency localization of Gabor functions (see Fig. 6.16).

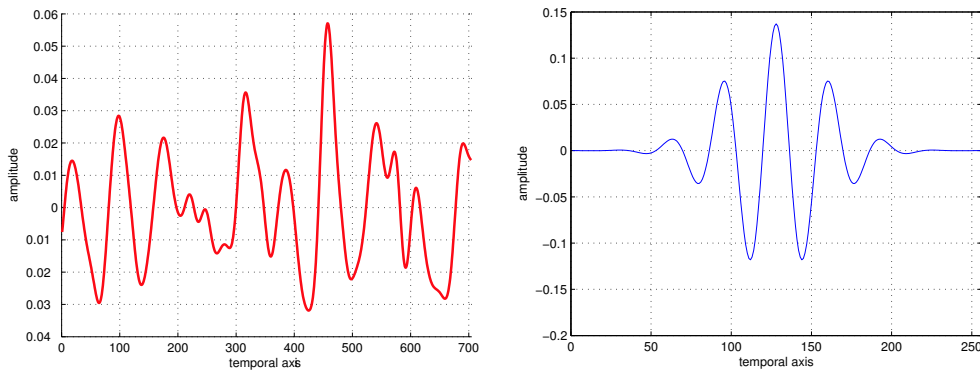


Figure 6.16: Left: Example of a simulated AA wave during fibrillation. Right: Gabor atom.

The dictionary we propose here is the result of empirical and theoretical studies on real and simulated ECG signals. Its design has been deeply studied in [125], where a particular care is reserved in bounding the coherence between \mathcal{D}_{AA} and \mathcal{D}_{VA} . Finally the MCD turns out to be highly redundant and able to well represent the ECG signals. However some problems are still present mainly concerning the T waves.

A priori information and sparse decomposition

Where does the *a priori* information come from? Mainly from medical knowledge about the position of the Q, R, S and T waves (see also [155]). In practice we also know that there are intervals where only AA is present, and it is easy to localize the QRS starting points.

Thanks to the structure of VA, f_{ECG} can be divided in VA periods. In addition, each VA period can be divided in a set of intervals corresponding to the different VA waves (Q, R, S and T) and an interval without ventricular activity. VA intervals can be estimated and identified in practice through the use of QRST point estimators. This information can thus be used to generate the weights of the dictionary atoms, i.e. the diagonal elements of the matrix W . The *a priori* knowledge obtained from [155] needs to be related with \mathcal{D} in the following way. \mathcal{D} is divided in \mathcal{D}_{AA} and \mathcal{D}_{VA} . Due to dynamics of AF, AA can be found through all the VA period. Hence, \mathcal{D}_{AA} atoms cannot be penalized. This is the reason why in this study we force: $w_l = 1 \quad \forall l : g_l \in \mathcal{D}_{AA}$. To the contrary, the selection of $g_l \in \mathcal{D}_{VA}$ can be successfully influenced by the use of the available *a priori* information. \mathcal{D}_{VA} is composed of a block optimized for QRS waves (ventricular depolarization) and a block designed for T waves (ventricular repolarization). Depending on the VA interval, w_l can be set to 1 for every $g_l \in \mathcal{D}_{VA}$ belonging to the appropriate kind for that interval. In case a g_l is unsuitable for a given interval, w_l can be set to a penalizing factor $0 \leq \tau < 1$. Thanks to the reliability of the estimators used in this work, it turned out that the best value for τ in our experiments is 0. For more details see [54, 125].

The signal is then decomposed by Weighted-Orthogonal Matching Pursuit. The weights stay constant through iterations, even if making them evolve can be interesting in order to better drive the iterative decomposition process.

Some Results

A biophysical computer model of the atria was used to obtain a realistic atrial electrical activity on the torso [101]. The AF signals that were generated in the 12-lead ECG were added to a clinical 4-second standard 12-lead ECG of an AF paroxysmal patient (78 years old) in sinus rhythm in which the P waves (AA) were removed. The clinical ECG was selected to represent the VA in AF as closely as possible. The ratio between the power of the original signal (simulated AA) and the estimation error (estimated AA - simulated AA) was used to evaluate the performance of our method.

First of all, we want to underline that we validated our choice of Weighted-OMP instead of OMP with these simulated measured 4-second ECG signals. By using Weighted-OMP, we increased the SNR in the recovery of VA (respectively, AA) by 0.81 dB (respectively, 0.65 dB). All the following results were obtained by approximating ECG signals with 50 atoms. However further research could be spent in order to select the optimal number of selected coefficients which should be, in general, signal dependent. Figure 6.17 shows the resulting separation of VA and AA of the simulated measured 4-second ECG signal on lead V1. One can see how the proposed method succeeds in approximating each one of the VA periods separating, at the same time, the AA with surprising fidelity. In order to study the influence of the AA amplitude on the method, three different simulated AA signals were created; 50, 100 and 150 % of the original simulated AA amplitude. The

ratio between the power of the original activity (VA or AA) and the error on the estimated one was evaluated on leads VR, V1 and V4 (see Table 6.1). We can observe that the quality of the AA estimation depends on the lead and its original amplitude. The AA SNRs are much higher with the 150% original amplitude and the overall performance on lead V1 is better than those of other two leads. Of course, the SNR values are directly related to the signal amplitude and in V1, the AA amplitude is higher compared to other leads. However, we observe a decrease of VA estimation performance in lead V1.

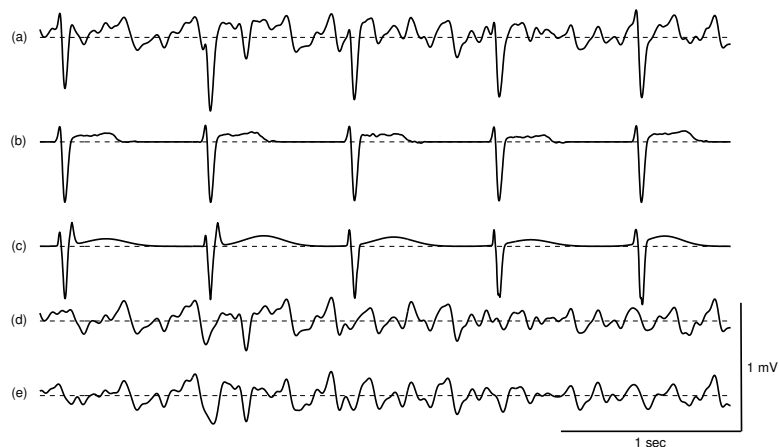


Figure 6.17: (a) Simulated measured 4-second ECG signal on V1. (b) Original VA on V1. (c) Estimated VA on V1 (SNR : 8.69 dB). (d) Simulated AA on V1. (e) Estimated AA on V1 (SNR : 6.81 dB)

Figure 6.18 shows the resulting separation of VA and AA of the clinical 4-second ECG signal on lead V2. These resulting signals have been validated using estimated power spectral densities (PSD). The dominant frequency of VA (respectively, AA) is between 1 and 2.5 Hz (respectively, between 3 and 10 Hz). The fact that there is no presence of VA dominant frequencies in the AA estimated PSD demonstrates the quality of our clinical results. For more experiments we refer again to [125].

| | 0.5·AA+VA | 1·AA+VA | 1.5·AA+VA |
|----------------|-----------|---------|-----------|
| lead VR | | | |
| VA SNR(dB) | 11.06 | 10.88 | 11.08 |
| AA SNR(dB) | -6.94 | -1.05 | 2.61 |
| lead V1 | | | |
| VA SNR(dB) | 11.13 | 8.69 | 2.41 |
| AA SNR(dB) | 3.61 | 6.81 | 4.28 |
| lead V4 | | | |
| VA SNR(dB) | 12.33 | 11.94 | 11.66 |
| AA SNR(dB) | -6.53 | -0.8 | 2.4 |

Table 6.1: Signal-to-noise ratio (dB) on lead VR, V1 and V4. The performance of our method is tested on 3 different AA amplitudes (50, 100 and 150 % of the original simulated signal).

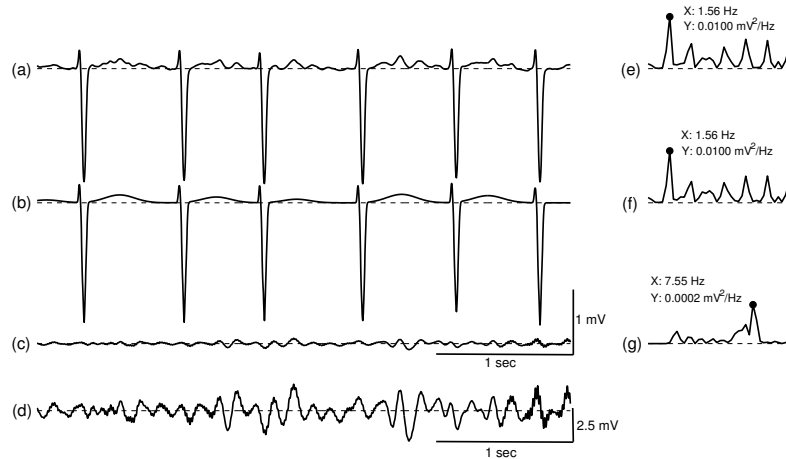


Figure 6.18: (a) Clinical 4-second ECG signal on V2 with a dominant frequency of 1.56 Hz (see its PSD (e)). (b) Estimated VA on V2 with a dominant frequency of 1.56 Hz (see its PSD (f)). (c) Estimated AA on V2 with a dominant frequency of 7.55 Hz (see its PSD (g)). (d) Estimated AA on V2 magnified 5 times.

6.7 Discussion

In this chapter we presented a method to take into account the information we have about the structure of a signal that we want to decompose. Many evidences prove that often the analysis algorithms fail in capturing the signal’s characteristics. Here we propose a solution to solve this problem. We already stressed how a natural signal is not only an array of numbers, but an expression of a physical event about which we usually have a deep knowledge. Therefore it is worth trying to exploit such knowledge, driving the decomposition to a better, more significant solution. Part of such knowledge is spent for the dictionary design, but usually it is not enough. In fact a dictionary, even if well suited for the class of data we want to work with, must be general and able to well represent all the members of that class. Here we propose a method to better link a very specific signal with the dictionary.

Remark that we use the expression *a priori* meaning “before the decomposition”. The information we refer to, comes from the very specific signal in use and can be extracted by a pre-analysis step, as shown in the examples. Of course the *a priori* information also depends on the dictionary. The utility of considering the *a priori* information is manifold and it can be particularly important when working with MCD. For another application we refer to [49].

The use of weights into some highly nonlinear approximation algorithms was previously suggested in [82]: in the FOCUSS algorithm (see Sec. 4.7.3) one can see the weights in W_{pk} as some kind of *a priori* knowledge about the solution. In this case an interpretation can be that the algorithm computes its own *a priori* information from iteration to iteration.

Another consideration concerns the algorithms based on QP, like Basis Pursuit Denoising and Weighted Basis Pursuit Denoising whose complexity is too high for many applications, as observed in Sec. 6.6.1. For this reason a greater applicability can be given by weighted greedy algorithms. In addition, it is easy to imagine including a weighting procedure in the thresholding algorithm. If the *a priori* information expressed by the weights is reliable one can claim that weighted thresholding is able to find good signal approximations, and of course at an extremely low computational cost.

Aside from complexity issues, personal experience and intuition suggest that the performances of

BPDN with huge and very redundant dictionaries like many of the ones presented in the examples are definitely worse than the ones of greedy algorithms. This in spite of the considerations about the myopia and the locality of greedy algorithms and, of course, independently of the use of weights. However, up to now there is no proof of this observation, but neither its contrary is verified.

Finally, we conclude this chapter saying how other forms of weighted algorithms are possible. for example in [85] we have used a weighted version of (P_{1-1}) , also solvable by Linear Programming methods.

Conclusion

A natural signal, even in its digital form, is not only an array of numbers, but an organized combination of mutually connected and dependent parts having, in general, a physical meaning. This dissertation investigates the problem of approximating and simplifying these signals taking advantage of their structure and, at the same time, preserving it.

The main contribution of this work is a decomposition framework based on multi-component redundant dictionaries, where any dictionary component is inspired by a signal characteristic. Such a concept depends on a predefined signal model and can be implemented either by using algorithms that take into account the *a priori* information available as shown in Chapter 6, or by dividing the decomposition in more steps, as done in Chapter 5 for natural images. In particular, the *a priori* information can appear under the form of weights: we propose weighted algorithms whose selection procedure is driven by the knowledge we have about the signal we are decomposing. This setting is studied from a theoretical and practical point of view, showing how important improvements upon the non-weighted case are achievable when reliable *priors* are available. Specifically, the class of redundant dictionaries for which it is possible to recover the sparsest approximation can be enlarged. Applications are presented for both 1-D and 2-D signals.

Secondly, a particular decomposition principle is proposed, consisting into a minimization of a cost function which is a trade-off between a sparsity measure and the error, both measured with the ℓ_1 norm. Its recovery properties are studied and examples show how it can be used for specific applications, such as denoising.

Moreover, this dissertation presents promising results in the area of still picture and video coding. For example, the algorithm for image compression described in Chapter 5 considerably outperforms the state of the art and it provides results that, up to my knowledge, none of the recently proposed coding technique is able to achieve. All these are evidences that the proposed framework offers strong and interesting potentialities.

In the Introduction, a random and a natural image appear in Figures 1.1 and 1.2. By the way the possible combinations of pixels that can be randomly generated are 2^{524288} , or, more clearly, $2^{256 \times 256 \times 8}$! Throughout the course of this dissertation we hope to have contributed to illustrate the difference between these two pictures and to develop a way of dealing with structured natural signals.

In short, we claim that when working with structured information, it is worth exploiting the mutual relation of its constituent elements, because it can be advantageous not only in helping

the analysis process but also in making the representation of such information more accessible and meaningful, determining its peculiar nature or character.

7.1 Possible Developments

One does not finish a thesis, s/he stops it! Thus, now I have in my mind very many possible developments and the list of future works can be long. Far from intending to further extend this work, I will anyhow briefly suggest some, in my opinion, interesting evolutions.

A good test for the proposed framework should be its application to other kinds of structured signals. Given its flexibility it can be fruitful provided that a signal model is available, a MCD can be designed and in case *a priori* weights can be computed. Specifically, one can think about standard applications in signal processing, as audio signals, rather than quite novel fields as computational linguistics.

When working with huge signals and dictionaries, the complexity of a decomposition by Matching Pursuit or Basis Pursuit (or equivalently their weighted versions) can be excessive. In these cases, if reliable *a priori* information is available, it can be interesting to study the approximation properties and performances of a Weighted Thresholding algorithm. It should be also possible to theoretically analyze its recovery capacities starting from the works in [52, 97].

To conclude, and this is really the right place to use this expression, one can observe that there is a deep link between sparse approximation and classification [80], and I believe that such relationship becomes even stronger when redundant MCD are in use. Decomposing a signal over a dictionary whose components reflect its constituent elements implies a strong similarity with the task of feature extraction and classification. Moreover, in the framework illustrated through this dissertation, this means to use meaningful features. It can be interesting to further explore such connection, as well as the relationship that may exist with the general problem of dimensionality reduction.

Appendix A

Two Proofs

A.1 Proof of Lemma 6.2

In this section we prove Lemma 6.2.

Proof: One can observe that solving (P_1^w) is equivalent to minimize the following function over coefficient vectors from \mathbb{R}^Λ :

$$F(\mathbf{b}) = \frac{1}{2} \|f_\Lambda - D_\Lambda \mathbf{b}\|_2^2 + \gamma \|W_\Lambda^{-1} \mathbf{b}\|_1.$$

A point \mathbf{b}_* minimizes the second term of $F(\mathbf{b})$ if and only if the following Fermat criterion holds (see [71, 109]): $\mathbf{0} \in \partial \|\mathbf{b}_*\|_1$. Moreover, \mathbf{b}_* minimizes $F(\mathbf{b})$ if and only if $\mathbf{0} \in \partial F(\mathbf{b}_*)$. In our case this means that

$$\exists \mathbf{u} \in \partial \|\mathbf{b}_*\|_1 \quad \text{s.t.} \quad D_\Lambda^T D_\Lambda \mathbf{b}_* - D_\Lambda^T f_\Lambda + \gamma W_\Lambda^{-1} \mathbf{u} = 0, \quad (\text{A.1})$$

for some vector \mathbf{u} taken from $\partial \|\mathbf{b}_*\|_1$. Let the atoms in Λ be linearly independent, from (A.1) it follows:

$$\mathbf{b}_* - (D_\Lambda^T D_\Lambda)^{-1} D_\Lambda^T f_\Lambda + \gamma (D_\Lambda^T D_\Lambda)^{-1} W_\Lambda^{-1} \mathbf{u} = 0,$$

and so

$$D_\Lambda^+ f_\Lambda - \mathbf{b}_* = \gamma (W_\Lambda D_\Lambda^T D_\Lambda)^{-1} \mathbf{u}.$$

To conclude the proof it is sufficient to recall that $\mathbf{c}_\Lambda = D_\Lambda^+ f_\Lambda$. ■

A.2 Proof of Corollary 6.1

We report here the proof of Corollary 6.1. We also refer to [50].

Proof: We want to prove that

$$\frac{\tau}{1 - \mu_1(m) - \mu_1(m-1)} \geq \frac{\tau \cdot \frac{w_\Gamma^{max}}{w_\Gamma^{min}} \cdot (1 - \mu_1^w(m-1) - \epsilon_{max})}{(1 - \epsilon_{max} - \mu_1^w(m) - \mu_1^w(m-1))(1 - \mu_1(m-1))}, \quad (\text{A.2})$$

If the *a priori* is reliable the weights corresponding to the atoms indexed by Γ are big. We have made the hypothesis that $\mu_1^w(m-1) + \mu_1^w(m) + \epsilon_{max} < \mu_1(m-1) + \mu_1(m) < 1$, from which we can assume:

$$\mu_1^w(m-1) + \mu_1^w(m) + \epsilon_{max} \leq \mu_1(m-1) + \mu_1(m) < 1, \quad (\text{A.3})$$

$$\mu_1^w(m-1) + \epsilon_{max} \leq \mu_1(m-1). \quad (\text{A.4})$$

Let us write (A.2) as:

$$1 \leq \frac{1 - \mu_1^w(m) - \mu_1^w(m-1) - \epsilon_{max}}{1 - \mu_1(m) - \mu_1(m-1)} \cdot \frac{1 - \mu_1(m-1)}{1 - \mu_1^w(m-1) - \epsilon_{max}} \cdot \frac{w_\Gamma^{min}}{w_\Gamma^{max}}, \quad (\text{A.5})$$

Analyzing the right-hand side of the previous expression we see that the third term is bigger than or equal to 1 by hypothesis. One just have to prove that

$$\frac{1 - \mu_1^w(m-1) - \epsilon_{max}}{1 - \mu_1(m-1)} \leq \frac{1 - \mu_1^w(m) - \mu_1^w(m-1) - \epsilon_{max}}{1 - \mu_1(m) - \mu_1(m-1)},$$

which is true since $\mu_1^w(m) \leq \mu_1(m)$. Therefore the Corollary 6.1 is proved. ■

Bibliography

- [1] (1992). *Information technology - digital compression and coding of continuous-tone still images - Part 1: requirements and guidelines. ISO/IEC International Standard 10918-1, ITU-T Rec. T.81.* ISO/IEC.
- [2] (2004). *ISO/IEC 15444-1:2004: JPEG 2000 image coding system: Core coding system.* ISO/IEC.
- [3] (2004). *ISO/IEC 15444-2:2004: JPEG 2000 image coding system: Extensions.* ISO/IEC.
- [4] (2005). NTT DoCoMo achieves 1Gbps packet transmission in 4G field experiment. NTT DoCoMo Press Release Article.
- [5] (2005). *Oxford English Dictionary.* Oxford University Press, third edn.
- [6] (URL). Bandelet web site. <http://www.cmap.polytechnique.fr/~peyre/bandelets/>.
- [7] (URL). Curvelet web site. <http://www.curvelet.org/>.
- [8] (URL). Java implementation of JPEG2000. <http://jpeg2000.epfl.ch/>.
- [9] (URL). JPEG2000 web site. <http://www.jpeg.org/jpeg2000/>.
- [10] (URL). NEOS. <http://www-fp.mcs.anl.gov/otc/Guide/>.
- [11] (URL). Netlib. <http://www.netlib.org/>.
- [12] (URL). USC-SIPI image database web site. <http://sipi.usc.edu/database/>.
- [13] (URL). Where is the starlet? http://lcd.siva.free.fr/where_is_the_starlet.html/.
- [14] S. Agarwal, A. Awan, D. Roth (2004). Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. Pattern Anal. Machine Intell.* **26**(11):1475–1490.
- [15] O. Al-Shaykh et al. (1998). Video compression using matching pursuits. *IEEE Trans. Circuits Syst. Video Technol.* **9**(1):113–143.
- [16] J.-P. Antoine et al. (2004). *Two-Dimensional Wavelets and their Relatives.* Cambridge University Press.
- [17] D. P. Bertsekas (1999). *Nonlinear Programming.* Athena Scientific, Belmont, Mass.
- [18] D. Bertsimas, J. N. Tsitsiklis (1997). *Introduction to Linear Optimization.* Athena Scientific, Belmont, Mass.

-
- [19] P. Bofill, M. Zibulevsky (2001). Underdetermined blind source separation using sparse representations. *Eurasip Signal Processing* **81**(11):2353–2362.
- [20] J. L. Borges (1941). *La biblioteca de Babel* from *El jardín de senderos que se bifurcan*.
- [21] L. Borup, M. Nielsen (2003). *Some remarks on shrinkage operators*. Tech. rep. Submitted.
- [22] R. W. Buccigrossi, E. Simoncelli (1999). Image compression via joint statistical characterization in the wavelet domain. *IEEE Trans. Image Processing* **8**(12):1688–1701.
- [23] P. J. Burt, E. H. Adelson (1983). The laplacian pyramid as a compact image code. *IEEE Trans. Commun.* **4**(31):532–540.
- [24] E. J. Candes, L. Demanet, D. L. Donoho, L. Ying (2005). *Fast Discrete Curvelet Transforms*. Tech. rep., Caltech, CA. Submitted.
- [25] E. J. Candes, D. L. Donoho (1999). Curvelets - a surprisingly effective nonadaptive representation for objects with edges. In C. A. Cohen, L.L. Schmaker (eds.), *Curve and Surface Fitting: Saint Malo 1999*, Vanderbilt University Press., Nashville, TN.
- [26] E. J. Candes, D. L. Donoho (1999). Ridgelets: a key to higher-dimensional intermittency? *Phil. Trans. R. Soc. Lond. A.* **357**:2495–2509.
- [27] E. J. Candes, D. L. Donoho (2003). New tight frames of curvelets and optimal representation of objects with piecewise C^2 singularities. *Pure and Applied Mathematics, Communication on* pp. 219–266.
- [28] E. J. Candes, M. Rudelson, T. Tao, R. Vershynin (2005). Error correction via linear programming. In *In Proc. of 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS05)*, IEEE.
- [29] E. J. Candes, T. Tao (2004). *Decoding by Linear Programming*. Tech. rep., Caltech, CA. Submitted.
- [30] J. Canny (1986). A computational approach to edge detection. *IEEE Trans. Pattern Anal. Machine Intell.* **8**(6):679–698.
- [31] F. Castells et al. (2003). Atrial fibrillation analysis based on ICA including statistical and temporal source information. In *Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP03)*, pp. V93–V96.
- [32] T. F. Chan, S. Esedoglu (2004). *Aspects of total variation regularized L^1 function approximation*. Tech. Rep. CAM Report 04-07, UCLA. To appear in *SIAM J. Appl. Math.*
- [33] S. S. Chen (1995). *Basis Pursuit*. Ph.D. thesis, Stanford University, Stanford, CA.
- [34] S. S. Chen, D. L. Donoho, M. A. Saunders (1999). Atomic decomposition by basis pursuit. *SIAM J. Sci. Comp.* **20**(1):33–61.
- [35] A. Cohen, I. Daubechies, J. C. Feauveau (1992). Biorthogonal bases of compactly supported wavelets. *Commun. Pure Appl. Math.* **45**:485–560.
- [36] A. Cohen, I. Daubechies, O. Guleryuz, M. Orchard (2002). On the importance of combining wavelet-based non-linear approximation with coding strategies. *IEEE Trans. Inform. Theory* **48**(7):1895–1921.

-
- [37] R. R. Coifman, M. V. Wickerhauser (1992). Entropy based algorithms for best basis selection. *IEEE Trans. Inform. Theory* **32**:712–718.
- [38] W. Czaja, M. V. Wickerhauser (2002). Singularity detection in images using dual local auto-covariance. *Applied and Computational Harmonic Analysis* **13**(1):77–88.
- [39] I. Daubechies (1988). Time-frequency localization operators: a geometric phase space approach. *IEEE Trans. Inform. Theory* **34**:605–612.
- [40] L. Daudet, S. Molla, B. Torresani (2004). Towards a hybrid audio coder. In *Proc. of Int. Conf. on Wavelet Analysis and its Applications (ICWAA04)*, pp. 13–24, World Scientific.
- [41] J. Daugman (1980). Two-dimensional spectral analysis of cortical receptive fields profile. *Vision Research* **20**:847–856.
- [42] M. Davies, L. Daudet (2003). Sparsifying subband decompositions. In *Proc. of Applications of Signal Processing to Audio and Acoustic, Workshop on*, IEEE.
- [43] M. Davies, L. Daudet (2004). Fast sparse subband decomposition using FIRSP. In *Proc. of the XII European Signal Processing Conference EUSIPCO 04*, EURASIP.
- [44] G. Davis, S. Mallat, M. Avellaneda (1997). Greedy adaptive approximation. *Journal of Constructive Approximation* **13**:57–98.
- [45] C. De Vleeschouwer, A. Zakhor (2002). Atom modulus quantization for matching pursuit video coding. In *Image Processing, International Conference on. ICIP02, Proceedings*, vol. 3, pp. 681–684, IEEE.
- [46] C. De Vleeschouwer, A. Zakhor (2003). In-loop atom modulus quantization for matching pursuit and its application to video coding. *IEEE Trans. Image Processing* **12**(10):1226–1242.
- [47] R. DeVore (1998). Non-linear approximation. *Acta Numerica* pp. 51–150.
- [48] R. DeVore, V. Temlyakov (1996). Some remarks on greedy algorithms. *Advances in Computational Mathematics* **5**:173–187.
- [49] O. Divorra Escoda (2005). *Toward Sparse and Geometry Adapted Video Approximations*. Ph.D. thesis, EPFL, 1015 Lausanne, Switzerland.
- [50] O. Divorra Escoda, L. Granai, P. Vandergheynst (2004). *On the Use of a priori Information for Sparse Signal Approximations*. ITS Technical Report 23.2004, ITS/LTS-2 EPFL, 1015 Lausanne, Switzerland.
- [51] O. Divorra Escoda, L. Granai, P. Vandergheynst (2004). *On the Use of a priori Information for Sparse Signal Representations*. ITS Technical Report 18.2004, ITS/LTS-2 EPFL, 1015 Lausanne, Switzerland.
- [52] O. Divorra Escoda, L. Granai, P. Vandergheynst (2005). On the use of *a priori* information for sparse signal approximations. *IEEE Trans. Signal Processing* To Appear.
- [53] O. Divorra Escoda, P. Vandergheynst (2003). Video coding using a deformation compensation algorithm based on adaptive matching pursuit image decompositions. In *Proc. of Int. Conf. on Image Processing (ICIP03)*, IEEE.

-
- [54] O. Divorra Escoda et al. (2005). Ventricular and atrial activity estimation through sparse ECG signal decomposition. Submitted.
- [55] M. N. Do, P. L. Dragotti, R. Shukla, M. Vetterli (2001). On the compression of 2-D piecewise smooth functions. In *Proc. of Int. Conf. on Image Processing (ICIP01)*, IEEE.
- [56] M. N. Do, M. Vetterli (2003). The contourlet transform: An efficient directional multiresolution image representation. *IEEE Trans. Image Processing* To appear.
- [57] D. L. Donoho (1998). Sparse components analysis and optimal atomic decomposition. *Constructive Approximation* **17**:353–382.
- [58] D. L. Donoho (1999). Wedgelets: nearly minimax estimation of edges. *Ann. Statist.* **27**:859–897.
- [59] D. L. Donoho, M. Elad (2003). Maximal sparsity representation via ℓ_1 minimization. *Proc. Nat. Acad. Sci.* **100**:2197–2202.
- [60] D. L. Donoho, M. Elad, V. Temlyakov (2005). Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Trans. Inform. Theory* To Appear.
- [61] D. L. Donoho, X. Huo (2001). Uncertainty principles and ideal atom decomposition. *IEEE Trans. Inform. Theory* **47**(7):2845–2862.
- [62] D. L. Donoho, I. M. Johnstone (1992). *Minimax estimation via wavelet shrinkage*. Tech. rep., Stanford University, Stanford, CA.
- [63] D. L. Donoho, M. Vetterli, R. DeVore, I. Daubechies (1998). Data compression and harmonic analysis. *IEEE Trans. Inform. Theory* **44**(6):2435–2476.
- [64] P. L. Dragotti, M. Vetterli (2003). Wavelet footprints: Theory, algorithms and applications. *IEEE Trans. Signal Processing* **51**(5):1306–1323.
- [65] M. Elad (2005). Why simple shrinkage is still relevant for redundant representations? *IEEE Trans. Inform. Theory* Submitted.
- [66] M. Elad, A. M. Bruckstein (2002). A generalized uncertainty principles and sparse representation in pairs of bases. *IEEE Trans. Inform. Theory* **48**(9):2558–2567.
- [67] R. M. Figueras i Ventura, L. Granai, P. Vandergheynst (2002). *A generalized Rate-Distortion limit for edge representation*. ITS Technical Report 07.02, ITS/LTS2 EPFL, 1015 Lausanne, Switzerland.
- [68] R. M. Figueras i Ventura, L. Granai, P. Vandergheynst (2002). R-D analysis of adaptive edge representations. In *Multimedia Signal Processing, Workshop on. MMSP02*, pp. 130–133, IEEE.
- [69] R. M. Figueras i Ventura, P. Vandergheynst, P. Frossard (2006). Low rate and flexible image coding with redundant representations. *IEEE Trans. Image Processing* To Appear.
- [70] R. M. Figueras i Ventura, P. Vandergheynst, P. Frossard, A. Cavallaro (2004). Color image scalable coding with matching pursuit. In *Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing, (ICASSP04)*, vol. 3, pp. 53–56, IEEE.
- [71] R. Fletcher (1997). *Practical Methods of Optimization*. Wiley-Interscience.

-
- [72] M. Frigo, S. Johnson (1998). FFTW: an adaptive software architecture for the FFT. In *Proc. of Int. Conf. on Acoustic, Speech and Signal Processing (ICASSP98)*, vol. 3, pp. 1381–1384, IEEE.
- [73] P. Frossard, P. Vandergheynst (2001). Redundancy in non-orthogonal transforms. In *Proc. of the Int. Symposium on Information Theory (ISIT01)*, p. 196, IEEE, Washington DC, USA.
- [74] P. Frossard, P. Vandergheynst, R. M. Figueras i Ventura (2003). High flexibility scalable image coding. In *Proc. of VCIP03*, SPIE.
- [75] P. Frossard, P. Vandergheynst, R. M. Figueras i Ventura, M. Kunt (2004). A posteriori quantization of progressive matching pursuit streams. *IEEE Trans. Signal Processing* **52**(2):525–535.
- [76] H. Fu, M. Ng, M. Nikolova, J. Barlow (2005). Efficient minimization methods of mixed L1-L1 and L2-L1 norms for image restoration. *SIAM Journal on Scientific Computing* To Appear.
- [77] J. J. Fuchs (2004). On sparse representations in arbitrary redundant bases. *IEEE Trans. Inform. Theory* **50**(6).
- [78] J. J. Fuchs (2005). Sparsity and uniqueness for some specific under-determined linear systems. In *Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP05)*, vol. 5, pp. 729–732, IEEE.
- [79] M. Ghanbari (2003). *Standard Codecs: Image Compression to Advanced Video Coding*. IEE, London.
- [80] F. Girosi (1998). An equivalence between sparse approximation and support vector machines. *Neural Computation* **10**(6):1455–1480.
- [81] I. F. Gorodnitsky, J. S. George, B. D. Rao (1995). Neuromagnetic source imaging with FOCUSS: A recursive weighted minimum norm algorithm. *J. Electroenceph. Clinical Neurophysiol.* **95**(4):231–251.
- [82] I. F. Gorodnitsky, B. D. Rao (1997). Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. *IEEE Trans. Signal Processing* **45**(3):600–616.
- [83] I. S. Gradshteyn, I. M. Ryzhik (1994). *Table of Integrals, Series, and Products*. Academic Press, Inc., fifth edn.
- [84] L. Granai, E. Maggio, L. Peotta, P. Vandergheynst (2004). Hybrid video coding based on bidimensional matching pursuit. *EURASIP Journal on Applied Signal Processing* **2004**(17):2705–2714.
- [85] L. Granai, P. Vandergheynst (2004). Sparse decomposition over multi-component redundant dictionaries. In *Proc. of Multimedia Signal Processing, Workshop on (MMSP04)*, pp. 494–497, IEEE.
- [86] L. Granai, P. Vandergheynst (2005). Sparse approximation by linear programming using an L1 data-fidelity term. In *Signal Processing with Adaptive Sparse Structured Representations (SPARS05), Workshop on*.
- [87] R. Gribonval (2002). Sparse decomposition of stereo signals with matching pursuit and application to blind separation of more than two sources from a stereo mixture. In *Proc. of Int. Conf. on Acoust. Speech Signal Process. (ICASSP02)*, IEEE, Orlando, Florida, USA.

-
- [88] R. Gribonval (2003). Piecewise linear source separation. In *Proc. of 48th SPIE annual meeting. Wavelets: Applications in Signal and Image Processing X*, vol. 5207, pp. 297–310, SPIE, San Diego, CA, USA.
- [89] R. Gribonval (2003). Piecewise linear source separation. Presentation at 48th SPIE annual meeting. *Wavelets: Applications in Signal and Image Processing X*.
- [90] R. Gribonval, E. Bacry (2003). Harmonic decomposition of audio signals with matching pursuit. *IEEE Trans. Signal Processing* **51**(1):101–111.
- [91] R. Gribonval, R. M. Figueras i Ventura, P. Vandergheynst (2006). A simple test to check the optimality of a sparse signal approximation. *EURASIP Signal Processing Journal* To Appear.
- [92] R. Gribonval, M. Nielsen (2003). Approximation with highly redundant dictionaries. In *Proc. of 48th SPIE annual meeting. Wavelets: Applications in Signal and Image Processing X*, SPIE, San Diego, CA, USA.
- [93] R. Gribonval, M. Nielsen (2003). *Highly sparse representations from dictionaries are unique and independent of the sparseness measure*. Tech. Rep. R-2003-16, Dept of Math. Sciences, Aalborg University.
- [94] R. Gribonval, M. Nielsen (2003). Sparse representations in unions of bases. *IEEE Trans. Inform. Theory* **49**(12):3320–3325.
- [95] R. Gribonval, M. Nielsen (2004). Nonlinear approximation with dictionaries. I. Direct estimates. *J. of Fourier Anal. and Appl.* **10**(1).
- [96] R. Gribonval, M. Nielsen (2006). Nonlinear approximation with dictionaries. II. Inverse estimates. *Constructive Approximation* To appear.
- [97] R. Gribonval, M. Nielsen, P. Vandergheynst (2005). *Stable recovery of structured approximations: a priori and a posteriori analysis*. Tech. rep. In preparation.
- [98] R. Gribonval, P. Vandergheynst (2006). On the exponential convergence of matching pursuit in quasi-incoherent dictionaries. *IEEE Trans. Inform. Theory* To Appear.
- [99] R. A. Horn, C. R. Johnson (1985). *Matrix Analysis*. Cambridge University Press.
- [100] X. Huo, J. Chen (2005). JBEAM: Multiscale curve coding via beamlets. *IEEE Trans. Image Processing* **14**(11):1665–1677.
- [101] V. Jacquemet et al. (2005). A biophysical model of eeg signals during atrial fibrillation used to evaluate the performance of QRST cancellation algorithms. In *Computers in Cardiology*, IEEE.
- [102] S. Jaggi, W. C. Karl, S. Mallat, A. S. Willsky (1998). High resolution pursuit for feature extraction. *Applied and Computational Harmonic Analysis* **5**:428–449.
- [103] L. K. Jones (1987). On a conjecture of huber concerning the convergence of projection pursuit regression. *Ann. Statist.* **15**(2):880–882.
- [104] P. Jost, P. Vandergheynst, P. Frossard (2004). *Tree-Based Pursuit*. Tech. Rep. 13.2005, ITS/LTS2 EPFL, 1015 Lausanne, Switzerland.

-
- [105] P. Jost, P. Vandergheynst, S. Lesage, R. Gribonval (2005). Matching of time invariant filters. In *Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS05)*.
- [106] B. Julesz (1962). Visual pattern discrimination. *IRE Tran. Information Theory* **IT-8**(1):84–92.
- [107] J. Kovacevic, M. Vetterli (1992). Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for R^n . *IEEE Trans. Inform. Theory* **38**:533–555.
- [108] S. Krstulovic, R. Gribonval, P. Leveau, L. Daudet (2005). A comparison of two extensions of the matching pursuit algorithm for the harmonic decomposition of sounds. In *Proc. of Workshop on Applications of Signal Processing to Audio and Acoustics*, IEEE.
- [109] A. Kusraev, S. Kutateladze (1995). *Subdifferentials: Theory and Applications*. Kluwer Academic Publishers.
- [110] E. Le Pennec, S. Mallat (2005). Sparse geometric image representations with bandelets. *IEEE Trans. Image Processing* **14**(4):423–438.
- [111] M. Lemay, J.-M. Vesin, Z. Ihara, L. Kappenberger (2004). Suppression of ventricular activity in the surface electrocardiogram of atrial fibrillation. In *ICA 2005*, pp. 1095–1102.
- [112] M. Lemay et al. (2005). Spatiotemporal QRST cancellation method using separate qrs and t-waves templates. In *Computers in Cardiology*, IEEE.
- [113] M. S. Lewicki, T. J. Sejnowski (2000). Learning overcomplete representations. *Neural Comp.* **12**:337–365.
- [114] J. Lin, W. Hwang, S. Pei (2005). SNR scalability based on bitplane coding of matching pursuit atoms at low bit rates. *IEEE Trans. Circuits Syst. Video Technol.* **15**(1):3–14.
- [115] Y. Lu, M. N. Do (2003). CRISP-contourlets: a critically sampled directional multiresolution image representation. In *Proc. of 48th SPIE annual meeting. Wavelets: Applications in Signal and Image Processing X*, SPIE, San Diego, USA.
- [116] E. Maggio (2003). *Un nuovo schema di codifica video con matching pursuit basato su una compensazione del moto H.264 compatibile*. Master’s thesis, Università degli studi di Siena, Italy.
- [117] D. M. Malioutov, M. Çetin, A. S. Willsky (2004). Optimal sparse representations in general overcomplete bases. In *Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP04)*, vol. 2, pp. 793–796, IEEE.
- [118] S. Mallat (1998). *A Wavelet Tour of Signal Processing*. Academic Press.
- [119] S. Mallat, F. Falzon (1998). Analysis of low bit rate image transform coding. *IEEE Trans. Signal Processing* **46**(4):1027–1042.
- [120] S. Mallat, Z. Zhang (1993). Matching pursuit with time-frequency dictionary. *IEEE Trans. Signal Processing* **41**(12):3397–3415.
- [121] H. S. Malvar, A. Hallapuro, M. Karczewicz, L. Kerofsky (2003). Low-complexity transform and quantization in H.264/AVC. *IEEE Trans. Circuits Syst. Video Technol.* **13**:598–603.

-
- [122] R. Malvar (2004). Is there life after JPEG2000 and H.264? Plenary talk at the Picture Coding Symposium.
- [123] Y. Meyer (2001). *Oscillating patterns in image processing and nonlinear evolution equations*, vol. 22 of *University Lecture Series*. American Mathematical Society, Providence, RI. The fifteenth Dean Jacqueline B. Lewis memorial lectures.
- [124] A. Miller (2002). *Subset Selection in Regression*. Chapman and Hall/CRC.
- [125] J. Molinero Hernandez (2005). *Sparse Decompositions for Ventricular and Atrial Activity Separation*. Master's thesis, EPFL.
- [126] S. Molla, B. Torresani (2004). Determining local transientness of audio signals. *IEEE Signal Processing Lett.* **11**(7):625–628.
- [127] S. Molla, B. Torresani (2005). An hybrid audio scheme using hidden markov models of waveforms. *Applied and Computational Harmonic Analysis* To Appear.
- [128] G. Monaci, O. Divorra Escoda, P. Vandergheynst (2005). Analysis of multimodal signals using redundant representations. In *Proc. of Int. Conf. on Image Processing (ICIP05)*, IEEE.
- [129] N. Moretti (1984). Bianca.
- [130] F. Moschetti, L. Granai, P. Vandergheynst, P. Frossard (2002). New dictionary and fast atom searching method for matching pursuit representation of displaced frame difference. In *Proc. of Int. Conf. on Image Processing (ICIP02)*, vol. 3, pp. 685–688, IEEE.
- [131] D. Mumford, J. Shah (1985). Boundary detection by minimizing functionals. In *Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, IEEE.
- [132] B. K. Natarajan (1995). Sparse approximate solutions to linear systems. *SIAM J. Comput.* **24**(2):227–234.
- [133] R. Neff, A. Zakhor (1997). Very low bit-rate video coding based on matching pursuit. *IEEE Trans. Circuits Syst. Video Technol.* **7**(1):158–171.
- [134] R. Neff, A. Zakhor (2000). Modulus quantization for matching-pursuit video coding. *IEEE Trans. Circuits Syst. Video Technol.* **10**(6):895–912.
- [135] R. Neff, A. Zakhor (2002). Matching-pursuit video coding, part I: Dictionary approximation. *IEEE Trans. Circuits Syst. Video Technol.* **12**(1):13–26.
- [136] R. Neff, A. Zakhor (2002). Matching-pursuit video coding, part II: Operational models for rate and distortion. *IEEE Trans. Circuits Syst. Video Technol.* **12**(1):27–39.
- [137] M. Nikolova (2002). Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers. *SIAM Journal on Num. Analysis* **20**:965–994.
- [138] M. Nikolova (2004). A variational approach to remove outliers and impulse noise. *J. Math. Imaging Vis.* **20**(1-2):99–120.
- [139] B. A. Olshausen, D. J. Field (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research* **37**:3311–3325.
- [140] B. A. Olshausen, D. J. Field (2000). Vision and the coding of natural images. *American Scientist* **88**:238–245.

-
- [141] B. A. Olshausen, D. J. Field (2003). *Problems in Systems Neuroscience*, chap. What is the other 85% of V1 doing ? Sejnowski, T.J., van Hemmen, Oxford University Press edn.
- [142] B. A. Olshausen, D. J. Field (2004). Sparse coding of sensory inputs. *Current Opinion in Neurobiology* **14**:481–487.
- [143] A. Ortego, K. Ramchandran (1998). Rate-distortion methods for image and video compression. *IEEE Signal Processing Mag.* **15**(6):23–50.
- [144] Y. C. Pati, R. Rezaifar, P. S. Krishnaprasad (1993). Orthogonal matching pursuit: Recursive function approximations to wavelet decomposition. In *Proc. of 27-th Annual Asilomar Conference on Signals Systems and Computers*.
- [145] W. B. Pennebaker, J. L. Mitchell (1993). *JPEG: Still Image Compression Standard*. Van Nostrand Reinhold, New York.
- [146] L. Peotta, L. Granai, P. Vandergheynst (2003). Very low bit rate image coding using redundant dictionaries. In *Proc. of 48th SPIE annual meeting. Wavelets: Applications in Signal and Image Processing X*, vol. 5207, pp. 228–239, SPIE, San Diego, USA.
- [147] L. Peotta, L. Granai, P. Vandergheynst (2005). Image compression using an edge adapted redundant dictionary and wavelets. *EURASIP Signal Processing Journal* To Appear.
- [148] G. Peyre, S. Mallat (2005). Discrete bandelets with geometric orthogonal filters. In *Proc. of Int. Conf. on Image Processing (ICIP05)*, IEEE.
- [149] J. D. Pinter (1996). *Global Optimization in Action*. Nonconvex Optimization and its Applications. Kluwer Academic Publishers, Dordrecht, The Netherlands.
- [150] J. Portilla, E. P. Simoncelli (1999). Texture modeling and synthesis using joint statistics of complex wavelet coefficients. In *Statistical and Computational Theories of Vision, Workshop on*, IEEE.
- [151] J. Portilla, E. P. Simoncelli (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. *Int. Journal of Computer Vision* **40**(1):49–71.
- [152] R. Q. Quiroga et al. (2005). Invariant visual representation by single neurons in the human brain. *Nature* **435**:1102–1107.
- [153] M. Rabbani, R. Joshi (2002). An overview of the JPEG2000 still image compression standard. *EURASIP Signal Processing: Image Communication* **17**:3–48.
- [154] A. Rahmoune, P. Vandergheynst, P. Frossard (2005). The M-term pursuit for image representation and progressive compression. In *Proc. of the Int. Conf. on Image Processing (ICIP05)*, IEEE.
- [155] R. M. Rangayyan (2002). *Biomedical Signal Analysis : A Case-Study Approach*. John Wiley & Sons, Inc.
- [156] A. Rosenfeld, A. C. Kak (1982). *Digital Picture Processing*. Academic Press, Inc., Orlando, FL, USA.

-
- [157] L. I. Rudin, S. Osher, E. Fatemi (1992). Nonlinear total variation based noise removal algorithms. In *Proc. of the Eleventh Annual Int. Conf. of the Center for Nonlinear Studies on Experimental mathematics: Computational issues in nonlinear science*, pp. 259–268, Elsevier North-Holland, Inc., Amsterdam, The Netherlands.
- [158] N. C. Rust, O. Schwartz, J. A. Movshon, E. P. Simoncelli (2005). Spatiotemporal elements of macaque V1 receptive fields. *Neuron* **46**(6):945–956.
- [159] S. Sardy (1998). *Regularization Techniques for Linear Regression with a Large Set of Carriers*. Ph.D. thesis, Univ. Washington, Seattle.
- [160] S. Sardy, A. Bruce, P. Tseng (2000). Block coordinate relaxation methods for nonparametric wavelet denoising. *Journal of Computational and Graphical Statistics* **9**(2):361–379.
- [161] S. Sardy, P. Tseng, A. Bruce (2001). Robust wavelet denoising. *IEEE Trans. Signal Processing* **49**(6).
- [162] R. Shukla, P. L. Dragotti, M. N. Do, M. Vetterli (2005). Rate-distortion optimized tree structured compression algorithms for piecewise smooth images. *IEEE Trans. Image Processing* **14**:343–359.
- [163] L. Sjooblum (1972). Miss november. *Playboy Magazine* Playboy Enterprises, Inc.
- [164] J.-L. Starck, D. L. Donoho, E. J. Candes (2003). Astronomical image representation by the curvelet transform. *Astronomy and Astrophysics* **398**:785–800.
- [165] J.-L. Starck, D. L. Donoho, E. J. Candes (2004). Image decomposition via the combination of sparse representations and a variational approach. *IEEE Trans. Image Processing* Submitted to.
- [166] J.-L. Starck et al. (2005). Morphological component analysis. In *Proceedings of the SPIE conference wavelets*, vol. 5914.
- [167] M. Stridh, L. Sörnmo (2001). Spatiotemporal qrst cancellation techniques for analysis of atrial fibrillation. *IEEE Trans. Biomed. Eng.* **48**:105–111.
- [168] W. Sweldens (1996). The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.* **3**(2):186–200.
- [169] W. Sweldens (1997). The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.* **29**(2):511–546.
- [170] D. Taubman, M. Marcellin (2001). *JPEG2000: Image compression fundamentals, standards and practice*. Kluwer Academic Publishers, Boston.
- [171] V. N. Temlyakov (2000). Weak greedy algorithms. *Adv. Comput. Math* (2-3):213–227.
- [172] V. N. Temlyakov (2002). Nonlinear methods of approximation. *Foundations of Comp. Math.* **3**(1):33–107.
- [173] L. Torres, M. Kunt (eds.) (1996). *Video Coding*. Springer.
- [174] J. A. Tropp (2004). Greed is good : Algorithmic results for sparse approximation. *IEEE Trans. Inform. Theory* **50**(10):2231–2242.

-
- [175] J. A. Tropp (2004). *Just relax: Convex programming methods for subset selection and sparse approximation*. Tech. rep., Texas Institute for Computational Engineering and Sciences.
- [176] J. A. Tropp (2004). *Topics in Sparse Approximations*. Ph.D. thesis, University of Texas, Austin.
- [177] P. Vandergheynst, P. Frossard (2001). Efficient image representation by anisotropic refinement in matching pursuit. In *Proc. of Int. Conf. on Acoustic, Speech and Signal Processing (ICASSP01)*, vol. 3, pp. 1757–1760, IEEE, Salt Lake City, USA.
- [178] V. Velisavljevic, B. Beferull-Lozano, M. Vetterli, P. L. Dragotti (2005). Approximation power of directionlets. In *Proc. of Int. Conf. on Image Processing (ICIP05)*, IEEE.
- [179] M. Vetterli, J. Kovacevic (1995). *Wavelets and Subband Coding*. Prentice-Hall.
- [180] M. Wakin, J. Romberg, H. Choi, R. Baraniuk (2002). Rate-distortion optimized image compression using wedgelets. In *Proc. of Int. Conf. on Image Processing (ICIP02)*, IEEE.
- [181] M. Wakin, J. Romberg, H. Choi, R. Baraniuk (2005). Wavelet-domain approximation and compression of piecewise smooth images. *IEEE Trans. Image Processing* To Appear.
- [182] M. W. Wickerhauser, W. Czaja (2004). A simple nonlinear filter for edge detection in images. In *Proceedings of SPIE*, vol. 5439, pp. 24–31.
- [183] T. Wiegand, G. J. Sullivan, G. Bjontegaard, A. Luthra (2003). Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.* **13**:560–576.
- [184] Y. Yuan, A. Evans, D. Monro (2004). Low complexity separable matching pursuits. In *Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP04)*, vol. 3, pp. 725–728, IEEE.
- [185] Y. Zhang (1998). Solving large-scale linear programs bt interior-point methods under Matlab enviroment. *Optimization Methods and Software* **10**:1–31.
- [186] D. P. Zipes, J. Jalife (1995). *Cardiac Electrophysiology : From Cell to bedside*. W.B. Saunders Compagny.

Curriculum Vitae

PERSONAL DATA

Name: **Lorenzo Granai**
Date of Birth: February 18, 1975
Place of Birth: Siena, Italy
Citizenship: Italian
Marital Status: Single
Email: lorenzo.granai@epfl.ch
Web page: <http://lts2www.epfl.ch/~granai>

EDUCATION

2002 - 2005: EPFL, Lausanne, Switzerland
Research Assistant & PhD. student at the Signal Processing Institute
Topic: Image Representation and Approximation Theory
2001: EPFL, Lausanne, Switzerland
Master Thesis at the Signal Processing Institute
Topic: Video Coding with Matching Pursuit
1994 - 2001: University of Siena, Siena, Italy
Telecommunication Engineering (final mark 109/110)

WORKING EXPERIENCES

2002 - 2005: Research Assistant & PhD. student at the Signal Processing Institute, EPFL
June - August 2002: Internship at NTT DoCoMo, Inc., Japan as a guest researcher within the Multimedia Signal Processing Laboratory
January - September 2001: Stage at Visiowave SA, Ecublens, Switzerland: Eight months Stage for the research project of my Master Thesis
December 1999 - May 2000: Teacher of computer science and multimedia applications. Siena, Italy

POST-GRADUATE COURSES

December 2003: HASSIP School on non-linear and adaptive approximation. Marseille, France
August 2004: Course on Multi-Scale Geometry in Image Processing and Coding at IPAM, UCLA, Los Angeles, CA

LANGUAGES

English: Fluent
French: Fluent
Spanish: Good
Italian: Mother Tongue

INTERESTS

Cinema (I was a cinema operator projectionist in 1999-2000), Reading, Philosophy, Figurative Art (I had a personal exposition in 2005 in Lausanne), Traveling, Basketball.

Personal Publications

Journal Papers

Oscar Divorra Escoda, Lorenzo Granai and Pierre Vandergheynst, *On the use of a priori information for sparse signal approximations*. IEEE Trans. on Signal Processing. To appear. (2005).

Lorenzo Peotta, Lorenzo Granai and Pierre Vandergheynst, *Image compression using an edge adapted redundant dictionary and wavelets*. Eurasip Signal Processing Journal, special issue on Sparse Approximations in Signal and Image Processing. To appear. (2005).

Lorenzo Granai, Emilio Maggio, Lorenzo Peotta and Pierre Vandergheynst, *Hybrid video coding based on bidimensional matching pursuit*. Eurasip Journal on Applied Signal Processing, Vol. 2004, No. 17, pp. 2705-2714, Dec 2004.

Conference Papers

Oscar Divorra Escoda, Lorenzo Granai, Mathieu Lemay, Javier Molinero Hernandez, Pierre Vandergheynst and Jean-Marc Vesin, *Ventricular and Atrial Activity Estimation Through Sparse ECG Signal Decompositions*. Submitted to ICASSP06.

Lorenzo Granai and Pierre Vandergheynst, *Sparse approximation by linear programming using an L1 data-fidelity term*. In Proc. of Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS'05), 16-18 November 2005, Rennes, FR.

Lorenzo Granai and Pierre Vandergheynst, *Sparse decomposition over multi-component redundant dictionaries*. In Proc. of Workshop on Multimedia Signal Processing (MMSP04), pp. 494-497, IEEE, September 2004, Siena, IT.

Lorenzo Peotta, Lorenzo Granai and Pierre Vandergheynst, *Very low bit rate image coding using redundant dictionaries*. In Proc. of 48th SPIE annual meeting, Wavelets: Applications in Signal and Image Processing X, vol. 5207, pp. 228-239, SPIE, November 2003, San Diego, CA, USA.

Lorenzo Granai, Fulvio Moschetti and Pierre Vandergheynst, *Ridgelet transform applied to motion compensated images*. In Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP03), vol. 3, pp. 381-384, April 2003.

Rosa M. Figueras i Ventura, Lorenzo Granai and Pierre Vandergheynst, *R-D analysis of adaptive edge representations*. In Proceedings of Workshop on Multimedia Signal Processing (MMSP02), pp. 130-133, IEEE, December 2002, St. Thomas, US Virgin Islands.

Fulvio Moschetti, Lorenzo Granai, Pierre Vandergheynst and Pascal Frossard, *New dictionary and fast atom searching methods for matching pursuit representation of displaced frame difference*. In Proc. of Int. Conf. on Image Processing (ICIP02), vol. 3, pp. 685-688, IEEE, September 2002, Rochester, NY, USA.

Technical Reports

Lorenzo Granai and Pierre Vandergheynst, *Sparse Approximation by linear programming: measuring the error with the ℓ_1 norm*. June 2005, Technical Report TR-ITS-2005.015.

Oscar Divorra Escoda, Lorenzo Granai and Pierre Vandergheynst, *On the use of a priori information for sparse signal approximations*. November 2004, ITS Technical Report 23/2004.

Oscar Divorra Escoda, Lorenzo Granai and Pierre Vandergheynst, *On the use of a priori information for sparse signal representations*. September 2004, ITS Technical Report 18/2004.

Lorenzo Granai, Emilio Maggio, Lorenzo Peotta and Pierre Vandergheynst, *Hybrid video coding using bi-dimensional matching pursuit*. December 2003, ITS Technical Report 08/2003.

Lorenzo Granai, *Radon and ridgelet transform applied to motion compensated images*. October 2002, ITS Technical Report 10/2002.

Rosa M. Figueras i Ventura, Lorenzo Granai and Pierre Vandergheynst, *A generalized rate-distortion limit for edge representation*. May 2002, ITS Technical Report 07/2002.