

SPARSE IMAGE APPROXIMATION WITH APPLICATION TO FLEXIBLE IMAGE CODING

THÈSE N^o 3284 (2005)

PRÉSENTÉE À LA FACULTÉ DES 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

Rosa Maria FIGUERAS i VENTURA

Enginyera de telecomunicació, Universitat Politècnica de Catalunya, Barcelone, Espagne
et de nationalité espagnole

acceptée sur proposition du jury:

Prof. P. Vandergheynst, directeur de thèse
Prof. R. Baraniuk, rapporteur
Prof. P. Frossard, rapporteur
Prof. R. Gribonval, rapporteur
Dr S. Sardy, rapporteur

Lausanne, EPFL
2005

A l'Oscar...

i als meus Pares i Germans.

Acknowledgments

During this thesis I have met a lot of people who helped me or gave me very nice moments. I will try to enumerate all of them here, but it is almost sure that I will forget somebody. I apologize in advance to all those who think that should be named here and that I forgot. Please, think as if you were named here!

I would like to thank all my thesis jury: Prof. Murat Kunt, Prof. Richard Baraniuk, Prof. Pascal Frossard, Prof. Rémi Gribonval, Dr. Sylvain Sardy and Prof. Pierre Vanderghenst. Thank you for your careful reading of the report, your pertinent comments and your interesting questions.

Bien sure, je voudrais remercier mon directeur de thèse, Prof. Pierre Vanderghenst. Merci, Pierre, pour tous les conseils et pour la liberté que tu m'a laissé pour choisir à chaque moment la direction de recherche que me plaisait plus.

Je voudrais aussi remercier Prof. Kunt, pour la bonne ambiance qui regne dans le, tout au début, laboratoire, et maintenant, Institut. Merci pour tous que j'ai appris au sein de son Institut.

And my greatest thanks to the presentation testers, Òscar and Philippe. Thank you for all your fruitful comments that allowed me to improve my exam presentation.

Merci aussi à Pascal Frossard, pour tout le travail qu'on a fait ensemble, pour ses conseils et sa patience, et pour me pousser, en toutes les tâches qu'on a partagé, à arriver plus et plus loin.

Merci à Rémi Gribonval pour le mois passé à Rennes. Merci pour toutes les heures passées au tableau et les intéressantes dissertations mathématiques.

This report would not have been what it is without the comments of several "volunteer" readers. Many thanks to all of you, Elena, Xavier, Olivier S., Vlad, Andrea, Pascal, Pierre and Òscar. Special thanks to Xavier for his help with the French translation of the abstract!

Big big thanks to my office-mate, Vlad. Thank you for your patience, your friendship and your support (and for letting me the laptop when mine was broken!). Thanks for being so calm when I was stressed, and for all your help.

Many thanks to Elena as well, for her advices concerning colors, her detailed reading of the report... and her company, of course!

Merci aussi à Xavier, pour avoir partagé les premières années de thèse à côté de moi dans le laboratoire (et avoir souffert mes bonnes propriétés envers les ordinateurs!). Merci de sa patience, son bon humeur... et sa capacité de trouver en tout moment le mot qu'il faut.

Merci à toute la gent du ITS que j'ai croisé: Diego (que compartiste el despacho durante algunos meses conmigo y suportaste estoicamente las inundaciones), Vlad, Patricia, Xavier, Raphael, Andrea, Francesco, Julien, Olivier, Lorenzo, Simon, Meri, Ruth, Torsten, Elisa, Elena, Philippe, Ana, David, Zen, Javi, Julien, Gianluca, Nicolas, Lisa et tout le reste. Merci à tous pour la bonne ambiance et

les activités extra-officielles qu'on a fait ensemble. Merci à Marianne, à Fabienne et à Gilles pour son aide logistique, parfois si précieuse!

Thanks to all the motivated students that I had during this time. Thank you because without you this work would not have reached the same milestones. Special thanks to Cristian. Gràcies per la feina ben feta, i per la teva simpatia i bona predisposició.

Merci aussi à tous les amis que j'ai fait à Lausanne. Gràcies a tots els catalans de Lausanne (sobretot al clan Ferrando i companyia!), que han fet que m'enyorés menys. Merci aux amis de la montagne (Patricia, Raphaël, Olivier, Pierre, Andrea, Òscar, Albert, Àngela et tous les autres).

Merci à tout l'équip de waterpolo, c'était vraiment sympa (et dur, parfois!).

Many thanks for my neighbor. Thanks Elisa for all the conversations through the window, the Sunday walking and everything else. Thank you very much!

I moltíssimes gràcies a la meva família, pel seu suport i la seva calor. Unes gràcies molt especials als meus pares, per haver-me encoratjat a anar més lluny i donat ànims quan els necessitava. Gràcies també als meus germans, Jordi i Anna (què faria sense vosaltres!). I gràcies a tots els meus oncles i tietes i cosins. Tot i que estigui lluny i no us pugui veure tan sovint com voldria, esteu sempre als meus pensaments.

I unes gràcies molt especials per a l'Òscar, que ha estat al meu costat en els moments difícils i m'ha intentat sempre ajudar. Gràcies també per haver-me empès quan no tenia forces per continuar... I gràcies per tots els bons moments que m'has donat durant tots aquests anys...

Gràcies a tots!

Merci à tous!

Thanks to everybody!

Abstract

Natural images are often modeled through piecewise-smooth regions. Region edges, which correspond to the contours of the objects, become, in this model, the main information of the signal. Contours have the property of being smooth functions along the direction of the edge, and irregularities on the perpendicular direction. Modeling edges with the minimum possible number of terms is of key importance for numerous applications, such as image coding, segmentation or denoising. Standard separable basis fail to provide sparse enough representation of contours, due to the fact that this kind of basis do not see the regularity of edges. In order to be able to detect this regularity, a new method based on (possibly redundant) sets of basis functions able to capture the geometry of images is needed.

This thesis presents, in a first stage, a study about the features that basis functions should have in order to provide sparse representations of a piecewise-smooth image. This study emphasizes the need for edge-adapted basis functions, capable to accurately capture local orientation and anisotropic scaling of image structures. The need of different anisotropy degrees and orientations in the basis function set leads to the use of redundant dictionaries. However, redundant dictionaries have the inconvenience of giving no unique sparse image decompositions, and from all the possible decompositions of a signal in a redundant dictionary, just the sparsest is needed. There are several algorithms that allow to find sparse decompositions over redundant dictionaries, but most of these algorithms do not always guarantee that the optimal approximation has been recovered. To cope with this problem, a mathematical study about the properties of sparse approximations is performed. From this, a test to check whether a given sparse approximation is the sparsest is provided.

The second part of this thesis presents a novel image approximation scheme, based on the use of a redundant dictionary. This scheme allows to have a good approximation of an image with a number of terms much smaller than the dimension of the signal. This novel approximation scheme is based on a dictionary formed by a combination of anisotropically refined and rotated wavelet-like mother functions and Gaussians. An efficient Full Search Matching Pursuit algorithm to perform the image decomposition in such a dictionary is designed.

Finally, a geometric image coding scheme based on the image approximated over the anisotropic and rotated dictionary of basis functions is designed. The coding performances of this dictionary are studied. Coefficient quantization appears to be of crucial importance in the design of a Matching Pursuit based coding scheme. Thus, a quantization scheme for the MP coefficients has been designed, based on the theoretical energy upper bound of the MP algorithm and the empirical observations of the coefficient distribution and evolution. Thanks to this quantization, our image coder provides low to medium bit-rate image approximations, while it allows for on the fly resolution switching and several other affine image transformations to be performed directly in the transformed domain.

Version abrégée

Les images naturelles sont souvent modélisées par des régions lisses par morceaux. Dans ce modèle, les bords de ces régions, correspondants aux contours des objets, constituent l'information principale du signal. Ces contours possèdent la propriété d'être des fonctions lisses le long de la direction des bords et irrégulières dans la direction orthogonale. La modélisation des bords avec un nombre minimal de termes est d'une importance capitale pour de nombreuses applications, telles que le codage d'image, la segmentation ou le débruitage. Les bases séparables standards ne peuvent pas fournir des représentations assez parcimonieuses des contours, dû au fait que ce type de base ne "voit" pas la régularité des bords. Par conséquent, une nouvelle méthode, basée sur les ensembles redondants de fonctions de base, est nécessaire pour capturer la géométrie des images.

Cette thèse présente, dans une première étape, une étude sur les caractéristiques que les fonctions de base devraient posséder pour fournir des représentations parcimonieuses d'une image lisse par morceaux. Cette étude souligne le besoin de fonctions de base adaptées aux bords qui soient capables de capturer précisément l'orientation locale et la dilatation anisotrope des structures de l'image. Des orientations et des degrés d'anisotropie différents dans un ensemble de fonctions de base conduisent à l'utilisation de dictionnaires redondants. Cependant, l'inconvénient des dictionnaires redondants est de fournir des décompositions d'image non uniques, alors que, de toutes les décompositions possibles d'un signal dans un dictionnaire redondant, seul la plus parcimonieuse est nécessaire. Plusieurs algorithmes permettent de trouver des décompositions parcimonieuses sur des dictionnaires redondants, mais la plupart de ces algorithmes ne garantissent pas de trouver l'approximation optimale. Afin de résoudre ce problème, nous avons mené une étude sur les propriétés des approximations parcimonieuses. Cette étude établit un test pour vérifier si une approximation parcimonieuse donnée est la plus parcimonieuse possible.

La seconde partie de cette thèse présente un nouveau schéma d'approximation d'image, basé sur l'utilisation d'un dictionnaire redondant. Ce schéma permet d'avoir une bonne approximation d'une image avec un nombre de termes beaucoup plus petit que la dimension du signal. Ce nouveau schéma d'approximation est basé sur un dictionnaire formé par la combinaison de Gaussiennes et de fonctions mères, similaires à des ondelettes anisotropement dilatées et tournées. Nous avons défini un algorithme "Matching Pursuit" (MP) efficace de recherche complète de pour accomplir la décomposition d'image dans un tel dictionnaire.

Finalement, nous construisons un schéma de codage géométrique d'images basé sur l'image approximée dans le dictionnaire anisotrope et tourné de fonctions de base. Nous présentons également une étude détaillée sur le codage des indices de fonctions de base dans l'approximation d'image parcimonieuse. Dans cette étude, la quantification des coefficients apparaît être d'une importance cruciale dans la conception d'un schéma de codage basé sur MP. Par conséquent, nous avons étudié un schéma de quantification des coefficients de MP, basé sur la borne supérieure théorique d'énergie de l'algorithme et les observations empiriques de la distribution et de l'évolution des coefficients.

Grâce à cette quantification, notre codeur d'image produit des approximations d'image à bas et moyens débits, compétitifs avec l'état de l'art, tout en permettant de changer de résolution en un clin d'oeil et d'exécuter plusieurs autres transformations géométriques directement dans le domaine transformé.

Resum

Les imatges naturals es modelen, sovint, com a regions regulars per parts. Els límits d'aquestes regions, que corresponen al contorn dels objectes, esdevenen, en aquest model, la part del senyal que aporta més informació. Modelar aquesta informació amb el mínim nombre de termes possible pot ser útil per a un gran nombre d'aplicacions, com codificació d'imatges, segmentació o eliminació de soroll.

Aquesta tesi efectua, en un primer estadi, un estudi de les característiques que han de tenir les funcions base per poder representar de manera esparsa el model d'imatge regular per parts. Aquest estudi posa en relleu el fet que per tal d'obtenir representacions més esparses (amb el mínim nombre de termes possible) cal que les funcions base siguin arbitràriament anisotròpiques i puguin orientar-se en qualsevol direcció. El fet de tenir funcions base amb rotacions i anisotropia dóna representacions més esparses. La utilització de diccionaris redundants, però, té un inconvenient: un senyal no té una única representació possible. Com que només la representació que té un nombre més petit de termes la que ens interessa, i els algorismes que existeixen per a representació o aproximació de senyals en diccionaris altament redundants no sempre garanteixen haver trobat aquesta solució, aquesta tesi efectua un estudi sobre l'optimalitat d'una determinada aproximació (o sigui, que aquesta representació és la que dóna, per a un cert nombre de termes, la solució amb el mínim error). D'aquest estudi ha sorgit, com a resultat principal, un test que permet comprovar si l'aproximació d'un senyal en un diccionari redundant és l'òptima (aquella que té menys error d'aproximació per al nombre de termes utilitzat).

La segona part de la tesi presenta un nou esquema per aproximar imatges, basat en l'ús d'un diccionari redundant. Aquest esquema permet tenir una bona aproximació d'una imatge amb un nombre de termes molt més petit que la dimensió de la imatge. Aquest nou mètode d'aproximació d'imatges es basa en l'ús d'un diccionari redundant, format per onetes (wavelets) amb anisotropia i rotació i Gaussians. En aquesta segona part s'hi explica la implementació de l'Algorisme Ggolafre amb Cerca Exhaustiva (Full Search Matching Pursuit) per efectuar decomposicions de les imatges en aquest diccionari. Aquest algorisme golafre amb cerca exhaustiva s'executa en un temps relativament ràpid (comparat amb l'algorisme golafre estàndard) perquè aprofita certes característiques del diccionari, com la seva localització espacio-freqüencial.

Per acabar, en aquesta tesi es presenta el disseny d'un esquema de codificació d'imatges a partir de les aproximacions obtingudes amb el diccionari redundant dissenyat a partir de la transformació anisotròpica i la rotació d'una funció base mare. Per dissenyar aquest codificador, s'ha efectuat un estudi detallat de com cal codificar els índex de les funcions base que intervenen en l'aproximació obtinguda en el diccionari redundant amb l'Algorisme Golafre. S'ha observat que un dels punts crucials per a poder codificar imatges amb la decomposició obtinguda amb l'algorisme golafre és la quantificació dels coeficients de les funcions base. Això ha portat a estudiar un esquema de quantificació dels coeficients obtinguts amb l'Algorisme Golafre, basat en el límit superior teòric de

l'energia d'aquest coeficients i les observacions empíriques de la seva distribució i evolució. Gràcies a aquest esquema de quantificació, s'ha creat un codificador d'imatges que funciona per a baixa i mitjana taxa de bits i que permet canvis de la resolució de la imatge i altres transformacions geomètriques en el domini transformat.

Contents

List of Figures	xxvi
List of Tables	xxvii
List of Algorithms	xxix
Abbreviations and Symbols	xxxi
1 Introduction	1
1.1 Motivation	1
1.2 Main contributions of the thesis	4
1.3 Thesis outline	5
2 Image Models and Geometry	7
2.1 Introduction	7
2.2 Motivation	7
2.3 How does the Human Visual System see Images?	8
2.3.1 Signal reception	9
2.3.2 Information coding and transmission in the retina	11
2.3.3 The Lateral Geniculate Nucleus	12
2.3.4 Visual Cortex V1	12
2.3.5 Visual Cortex V2	14
2.3.6 V3 and above	15
2.3.7 Recognition of objects and masking effects	16
2.3.8 Why psychovisual effects are so difficult to take into account?	18
2.3.9 What can be imitated and to what point?	18
2.4 Image models	19
2.5 Introducing geometry in image representation	21
2.5.1 Horizon Model	22
2.5.2 Statistical model	22
2.5.3 Other models	23
2.5.4 Some remarks	24
2.6 Representation through Harmonic Analysis	24
2.6.1 From Fourier to Wavelets	24

2.6.2	Redundant representations	25
2.7	Some geometrical oriented representation techniques	26
2.7.1	Curvelets	26
2.7.2	Contourlets	27
2.7.3	Wedgelets	29
2.7.4	Footprints/Edgeprints	30
2.7.5	Bandelets	31
2.7.6	Some remarks	32
2.8	Conclusions	32
3	Towards geometric image representations	33
3.1	Introduction	33
3.2	Image model in use	34
3.3	Quadtree	34
3.3.1	Number of terms in the decomposition	35
3.3.2	Distortion of the decomposition	35
3.3.3	Rate-Distortion behavior	36
3.4	Optimized quadtree	36
3.4.1	Number of terms in the decomposition	36
3.4.2	Distortion	37
3.4.3	Rate-Distortion behavior	37
3.5	Anisotropic quadtree	37
3.5.1	Number of terms in the decomposition	37
3.5.2	Distortion	38
3.5.3	Rate-Distortion	39
3.6	Anisotropic quadtree with refinement	39
3.6.1	Number of terms	40
3.6.2	Distortion	40
3.6.3	Rate-Distortion	40
3.7	Anisotropic quadtree with rotation	42
3.7.1	Number of terms	42
3.7.2	Distortion	45
3.7.3	Rate-Distortion	46
3.8	Anisotropic quadtree with rotation and refinement	47
3.8.1	Number of terms	47
3.8.2	Distortion	47
3.8.3	Rate-Distortion	48
3.9	Practical results	49
3.10	Conclusions	50
4	Sparse signal representations	53
4.1	Introduction	53
4.2	Some definitions	54
4.3	Sparse signal representation with redundant dictionaries	55
4.3.1	Basis Pursuit and Basis Pursuit Denoising	55
4.3.2	Greedy algorithms	56
4.3.3	Weak Greedy algorithms	58
4.3.4	Other existing algorithms	58

4.4	Recoverability of sparse signal approximations	59
4.4.1	Basis Pursuit	59
4.4.2	Basis Pursuit Denosing	60
4.4.3	Matching Pursuit and Orthogonal Matching Pursuit	61
4.5	Conclusions	62
5	An algorithm independent sparsity measure	63
5.1	Introduction	63
5.2	Some remarks about the optimality test	64
5.3	Main concepts and results	65
5.3.1	Main theorems in a Hilbert space	67
5.3.2	What do these results mean?	69
5.3.3	Explicit tests of optimality in a Hilbert space	70
5.3.4	Examples	71
5.4	Core elements of the proofs	73
5.5	Estimates in the canonical basis	74
5.6	Sub-additive distortion measures	75
5.7	Proof of the main results	77
5.7.1	Proof of Theorem 5.1.	78
5.7.2	Proof of Theorem 5.2.	78
5.7.3	Proof of Proposition 5.1	80
5.7.4	Sharpness of Theorem 5.1	81
5.7.5	Sharpness of Theorem 5.2	81
5.8	Conclusions	81
6	Image approximation with Matching Pursuit	83
6.1	Image Model	84
6.2	Dictionary of basis functions	84
6.2.1	Generating functions	84
6.2.2	Anisotropy and orientation	85
6.2.3	Dictionary	85
6.2.4	Benefits of anisotropy	86
6.2.5	Representation of the low frequencies of the image	87
6.3	GA based MP: a Weak MP implementation	87
6.4	Full Search MP	89
6.4.1	“Brute Force” Full Search MP	90
6.4.2	Spatial invariance in scalar product computation and boundary renormalization	91
6.4.3	FFT based FS-MP: from Scalar Products to Spatial Convolution	93
6.4.4	Results: Full Search vs Genetic Algorithm Search	95
6.5	Exploiting the dictionary features	96
6.5.1	Exploiting Spatio-Temporal Energy Localization: Compact Support and Atoms Approximation	97
6.5.2	Steerability of Atoms and Complexity Benefits	101
6.6	MP with low-pass and band-pass dictionary	102
6.6.1	Construction of the dictionary	105
6.7	Fully flexible image representation with MP	106
6.8	MP with <i>a priori</i> knowledge: Weighted MP	108
6.8.1	Edge dictionary	109

6.8.2	Texture dictionary	109
6.8.3	Smooth parts dictionary	109
6.8.4	Comments about the final dictionary	109
6.8.5	Computation of the weights	110
6.8.6	Weighted MP image representation	116
6.9	Color images: multichannel MP	118
6.9.1	Vector MP	118
6.9.2	Color space choice	119
6.10	Conclusions	120
7	Flexible image coding with Matching Pursuit	123
7.1	Introduction	123
7.2	Reduction of the entropy of the function index	124
7.3	<i>A posteriori</i> MP coefficients quantization	125
7.3.1	Redundancy-Driven <i>a posteriori</i> MP coefficients quantization	126
7.3.2	Practical Scheme: Adaptive Quantization	134
7.3.3	Probability distribution optimized MP coefficient quantization	136
7.3.4	Reconstruction of MP Coefficients by Interpolation	139
7.4	Matching Pursuit Image Coder structure	143
7.4.1	Overview	143
7.4.2	First coding scheme	144
7.4.3	Enhanced coding and decoding scheme	145
7.5	Coding Performance	146
7.5.1	Benefits of anisotropy	146
7.5.2	Rate-distortion performance	146
7.5.3	Visual comparisons	147
7.5.4	Discussion	153
7.6	High adaptivity	157
7.6.1	Importance of adaptivity	157
7.6.2	Spatial adaptivity	158
7.6.3	Rate scalability	161
7.7	Color image coding with Matching Pursuit	163
7.7.1	Color Image Parameter coding	163
7.7.2	Results	164
7.8	Conclusions	167
8	Conclusions	169
8.1	Summary	169
8.2	Future research	171
A	Some proves on the algorithm independent sparsity measures	175
A.1	Sparsity measures	175
A.2	Proof of Lemma 5.1	176

B Computation of the sampling of the dictionary parameters	177
B.1 Dictionary \mathcal{D}_{AR} based on AR atoms	177
B.1.1 A Typical Parameter Discretization	179
B.2 Dictionary \mathcal{D}_G based on Gaussian atoms	179
B.2.1 A Typical Parameter Discretization	181
B.3 Dictionary \mathcal{D}_{Ga} based on Gabor Atoms	182
B.3.1 A typical parameter discretization	183
Bibliography	185
Curriculum Vitae	199
Publications	201

List of Figures

1.1	Wavelet versus edge adaptive scheme: Illustration of the fact that wavelets fail to detect the regularity of contours, while a set of basis functions that includes anisotropy and rotation would detect it.	2
1.2	Basis functions obtained with Olshausen and Field's method [139] from images of size 12×12 . The basis functions have been obtained by learning a sparse code from a set of natural images patches.	3
2.1	Human Visual System structure (a) shows the path the information follows, from the reception to the brain, in the HVS (Figure taken from [146]). (b) gives a schematic vision of the different layers of the HVS used for still image reception and processing, and a simplified version of the information transmission and processing. .	9
2.2	Information reception in the Human Visual System. Images are received in the Human eye (a), by projecting the light receiving trough the pupil to the retina, where there are different kinds of receptors (b).(Images taken from [102])	10
2.3	The blind spot. This little experiment shows empirically the existence of the blind spot. Close your left eye and stare at the cross mark in the diagram with your right eye. Off to the right you should be able to see the spot. Don't LOOK at it; just notice that it is there off to the right (if its not, move farther away). Now slowly move toward the page. Keep looking at the cross mark while you move. At a particular distance (probably around 30cm), the spot will disappear (it will reappear again if you move even closer). The spot disappears because it falls on the optic nerve head, the hole in the photoreceptor sheet. (Image and caption have been taken from [169].)	11
2.4	(a) Retina ganglion cells response, and its connections to bipolar cells for fovea and (b) for the rest non-foveal ganglion cells. (c) shows the overlap that can occur between neighboring ganglion cells, specially for those that are not in the fovea. (Images (a) and (b) taken from [102])	13
2.5	V1 cells response to several stimuli. Figure 2.5(a) shows the simple cells field and Figure 2.5(b) shows the complex cells field. The white stripe represents the stimulus received by the cell. The cell receptive field is represented in red when the stimulus excites the cell, and in blue when the cell does not respond to the stimulus.	14
2.6	Basis functions obtained with Olshausen and Field's method [139] from images of size 12×12	15

2.7	Some artistic images that profit from the HVS strong and weak points to create interesting visual effects. (a) takes profit from the ability of the HVS to detect faces to hide faces in the whole picture (some of them really evil). (b) exploits some of the lacks in information processing of the HVS to make the observer think that the picture has wavy lines, when in fact all the lines are perfectly straight. Both figures are taken from [1], where there are many other examples of this kind of visual effects.	16
2.8	More optical illusions from [1]. (a) shows the effect shadows have on the perception of the third dimension. This figure is interpreted by the eye as two rows of bumps and two of dimples (generally, shadows on the bottom of the spot make the observer interpret it as a bump, and shadows on the top make it be seen as a dimple). But careful! The third row is not exactly what it seems... This shows that the Human Brain considers the world as being regular, and that an irregularity is only perceived when the scene is watched more attentively. (b) shows an example of the capability of interpolation and interpretation the HVS has. Even though the dog is not perfectly drawn, it is just a set of spots, a Human observer interprets the scene and perfectly sees the dog.	17
2.9	Example of image with occlusions. The bicycle is occluding a part of the brick wall which is occluding the tree, which in its turn, is occluding the sun.	20
2.10	HVS versus machine perception of contours, lines and objects. (a) Example of the sampling of a straight line. The human eye easily perceives a straight line, even if it has some discontinuities. (b) Example of interpolation performed by the HVS. In spite of the occlusions, our brain is capable of seeing the B (thus, of interpolating edges where discontinuities are present).	21
2.11	The Horizon model and some of its variations.	23
2.12	Edge decomposition with different kinds of basis functions. This example shows that when using Isotropic basis functions (a), the number of functions needed to represent a curve with a certain error is higher than when using anisotropic and rotated basis functions (b).	25
2.13	Curvelets: frequency domain division. Curvelets divide the frequency domain both in angle and in frequency rings. The number of divisions that increases with the frequency.	27
2.14	Contourlets and Contourlet Packets Frequency domain division. Contourlets perform a parabolic scaling of the frequency domain, while contourlet packets perform an adaptive division, dependent on the signal to code (image taken from [41]).	28
2.15	Wedgelet representation of a cartoon image. Figure 2.15(a) shows the original image and Figure 2.15(b) is its wedgelet approximation, with a zoom in one leaf to show a wedgelet description in the quadtree leaf (image taken from [6]).	29
2.16	Footprint. Time domain (top) and wavelet domain (bottom) representation of the footprint $f_k^{(0)}$ with $N = 128$ and $k = 41$ (taken from [51]). Except for the case $J = \log_2 N$ (N being a power of 2), a footprint does look like a pure step edge, since it does not include the scaling coefficients (image taken from [51]).	31
3.1	Isotropic and Anisotropic quadtree. (a) Example of the behavior of the isotropic quadtree, or standard quadtree, when trying to approximate an edge. (b) Example of the behavior of the anisotropic quadtree when trying to approximate an edge.	35
3.2	Anisotropic quadtree with rotation. (a) shows how the decomposition at one resolution level behaves, and (b) shows the final decomposition results when using 4 decomposition levels.	42

3.3	Comparison among JPEG2000, isotropic quadtree with refinement and anisotropic quadtree with rotation for an image of 1024×1024 pixels.	49
3.4	Comparison of the bit-rate of different curves with different Total Variation.	50
6.1	An example of one atom of the dictionary in the spatial domain and in the frequency domain.	86
6.2	Comparison of the approximation properties of different dictionaries of functions: orthogonal Haar wavelet basis, bi-orthogonal Daubechies wavelet basis, non-oriented Gabor functions, oriented Gabor functions and oriented and anisotropically refined atoms.	87
6.3	Genetic Algorithm block diagram	89
6.4	Description of the setting up procedure to generate the look up tables to be used in Figure 6.5	94
6.5	Schematic description of the full search algorithm for one iteration of MP. Look up tables are used to hold the DFT of the dictionary functions and the normalizing masks.	95
6.6	Original <i>Lenna</i> , <i>Cameraman</i> and <i>Baboon</i> images.	96
6.7	Visual comparison of <i>Lenna</i> 128x128 decomposed with MP with 300 coefficients (a) with the GA-MP and (b) with the FS-MP.	96
6.8	Visual comparison of <i>Cameraman</i> 128x128 decomposed with MP with 300 coefficients (a) with the Genetic Algorithm and (b) with the Full Search MP.	97
6.9	Visual comparison of <i>Baboon</i> 128x128 decomposed with MP with 300 coefficients (a) with the Genetic Algorithm and (b) with the Full Search MP.	97
6.10	Evolution of the PSNR with the iteration number for the Genetic and the FS-MP algorithms. The image used in this case is the gray-scale 128x128 <i>Lenna</i>	98
6.11	Example of different functions in the dictionary and their respective normalization masks. (a) Frequency modulus of three selected dictionary atoms (white corresponds to high values and black to zero). (b) Respective support of the significant values that need to be stored in memory (in white, values with modulus greater than 0.001). (c) Spatial renormalization maps that correspond to the atoms of the first row (white corresponds to 1 and black to zero). (d) White zones represent the binary mask that determines where values different from 1.0 with a significant difference (greater than 0.001) are located.	100
6.12	Every Fourier domain atom is stored in a run-length fashion. All values considered non-significant (black zones in the figure) are set to zero. All consecutive zeros (considering a line wise scanning of images, as indicated by the red arrows) are efficiently stored using a single integer. Significant values (white zones in the image) are stored one by one together with an integer number that specifies how many significant values are consecutively aligned.	101
6.13	Approximation PSNR vs memory used for the FFT based full search algorithm for <i>Lenna</i> 128x128. In the left the approximation is done with 100 terms. In the right, 300 terms are used. The graphics have been obtained by increasing the threshold of approximation logarithmically. It can be seen that up to 75% in memory savings can be achieved without loss in approximation convergence.	102
6.14	Visual comparison of the different approximation of <i>Lenna</i> 128x128 for the amounts of memory of: (a) 377 MB (threshold 0), (b) 122 MB (threshold 10^{-4}), (c) 22 MB (threshold 1) and (d) 2.2 MB (threshold 10).	103

6.15	In some cases, a whole set of functions from the dictionary can be generated from the linear combination of a subset of these. This is the case, for example, for isotropically scaled Gaussian second derivatives [69], only three real filters (drawn in continuous line) are needed to generate all the rest of orientations.	104
6.16	Comparison of the growth of complexity with and without using steerability as a function of N . This graph has been performed assuming $N_\theta = 36$	104
6.17	Example of AR and Gaussian atoms and their respective Fourier Transforms.	105
6.18	Visual comparison of the image obtained with MP without the low pass with 500 terms (a) and MP with the low-pass dictionary with 500 terms(b), and evolution of the PSNR with the number of coefficients for <i>Lenna</i> 128×128 image.	107
6.19	Example of texture coding with different dictionaries [19]. Subfigure (a) corresponds to a detail of Barbara and (b) and (c) to the MP coding with 50 AR atoms and Gabor atoms respectively. It can be seen both through the visual comparison and through the PSNR plot that Gabor atoms capture more efficiently the textures than AR atoms.	110
6.20	PSNR for MP approximations using various dictionaries types over Barbara image. It is shown that the combined use of Anisotropic Refinement and Gabor dictionaries (AR GABOR) leads to better PSNR ratios than the use of single dictionaries (AR or GABOR) because it catches better the structures of the image.	111
6.21	Edge representation by two Anisotropic Refinement atoms. When MP represents an edge, assuming a simple model of two atoms, the result looks like this. The green line represents the edge and the two red spots the center of the two atoms. This figure shows that to shape a step edge, the atoms are not on the edge itself but near it. Nevertheless, not all the edges are step edges, and not all of them require of two AR atoms to be approximated.	112
6.22	Edge Model. The red plot models a step edge, while the green represents the simplest representation of such an edge by taking a linear combination of two Anisotropic Refinement atoms.	113
6.23	Atom placement area δ	114
6.24	Impulsional response of the edge model mask linear filter	114
6.25	Waterfilling. The α parameter allows to control the influence of the masks into the pondering process. In (a), the original a 1D-Mask. In (b), its waterfilled result with a high value of α and, in (c), with a lower value. Note that in (c) begins to appear a weak probability region (lower peak).	116
6.26	Structure of the Hybrid MP coder. It can be seen the Hybrid MP coder works similarly to the FS-MP coder, but with a pondering of the scalar product energy when in the choice of the basis functions.	117
6.27	Distribution of the MP coefficients when MP is performed in the RGB color space (left), with the diagonal of the cube shown in red, and in the YUV color space (right).	119
6.28	Original Japanese woman image	120
6.29	(a) Japanese woman coded with MP using the most energetic channel search strategy in YUV color space and (b) in RGB color space.	121
7.1	In loop and out loop image quantization schemes. The in loop MP quantization scheme subtracts to the original signal the quantized projection, while the out-loop uses the non-quantized values for image subtraction.	128

7.2	Comparison of <i>a priori</i> and <i>a posteriori</i> uniform coefficient quantization of the MP expansion of 10-sample random real signals over a dictionary of 128 random vectors. The distortion have been averaged over 10 independent encodings. To draw this graph, the quantization step size has been kept constant, and both the <i>a priori</i> and <i>a posteriori</i> quantization schemes have been applied the same quantizer. The increase of distortion at the higher rates comes from the fact that, for high iteration number, the coefficients become smaller than one half of the quantization step, and thus the distortion due to quantization is higher than the energy that these basis functions subtract.	129
7.3	Coefficient norm versus the iteration number for the MP expansion of 3 random signals of length 10 over a 128-random vectors dictionary ($\alpha = 1$).	130
7.4	Distribution of coefficients $n = \{0, 4, 9, 19\}$ relatively to the exponential upper-bound, for the MP expansion of thousand 10-sample random signals over a random dictionary of 128 atoms ($\alpha = 1$).	130
7.5	Average R-D curve for the MP expansion of ten 10-sample random real signals over a 50-vector random dictionary, in the case of both exponential and uniform quantization.	133
7.6	Rate-distortion curve for adaptive and exponential quantization the MP expansion of 10-sample random real signals over a 128-atom dictionary.	136
7.7	(a) Evolution of the mean of the log of the MP coefficients with the log of the iteration number. It can be seen that the behavior is nearly linear. The mean was estimated from 88 different images. (b) Evolution for a single image coefficient stream of the log of its coefficients with the log of the iteration number. It can be seen that for a single image, the behavior of the coefficients does not differ from the evolution of the mean.	137
7.8	(a) Solid line: Evolution of the variance of z_i with the iteration number. Dashed line: Evolution of the variance of ϖ with the iteration number. Dot-dashed line: Evolution of $i^2\sigma_\varpi^2$ with the iteration number.(b) Evolution of the kurtosis (solid line) and the skewness (dashed line) of z	138
7.9	(a) Evolution of the empirical mean value of ϖ with the iteration number. (b) Evolution of the value of ϖ for ten image coefficient streams.	139
7.10	Image <i>Goldhill</i> (256x256): Behavior of unquantified coefficient magnitudes in log-log scale compared to dequantized values using 1024 initial quantization levels.	140
7.11	Image <i>Barbara</i> (256x256): Approximation of coefficient magnitudes with Eq. (7.24); parameters a , b and ϖ are calculated by using a linear regression on original coefficient values for iteration numbers between 120 and 1000.	141
7.12	Selection of end-points A and B for interpolating from dequantized values in a given constant region; end points are denoted by asterisks.	142
7.13	Image <i>Cameraman</i> (256x256): interpolation with the rational function and 128 initial levels, compared to non-quantized values. Inappropriate interpolation (done before the number of quantization levels becomes low enough) is also shown.	142
7.14	Image <i>Cameraman</i> (256x256): Optimal number of levels for the first atom, for the given bit rate of 0.2bpp. The number of levels for the first atom ranges between 128 and 4096 with a resolution of 3 values per octave.	143
7.15	Block diagram of the Matching Pursuit image coder.	144

7.16	Comparison of the rate distortion of anisotropic refinement in the dictionary and its isotropic version. The basis functions used are the same for isotropic and anisotropic functions, with the same parameter discretization, showing the Rate-Distortion improvement of adding anisotropy to the dictionary.	147
7.17	Distortion-rate performance for JPEG2000, SPIHT and the proposed MP coder, for several test images.	148
7.18	<i>Lenna</i> 256 × 256 encoded at 0.35 bpp with MP, JPEG2000 and SPIHT.	150
7.19	<i>Lenna</i> (512 x 512) encoded with MP and JPEG2000 at 0.16 bpp.	152
7.20	Detail view of <i>Lenna</i> (512 × 512) encoded at 0.16 bpp.	153
7.21	<i>Goldhill</i> 256 × 256 encoded at 0.23 bpp.	154
7.22	<i>Barbara</i> (256 x 256) encoded at 0.12 bpp.	156
7.23	Image Cameraman (256x256): The RD performance obtained with the optimal initial number of levels from the Figure 7.14 (203 initial quantization levels with interpolation).	157
7.24	<i>Lenna</i> image of size 256 × 256 encoded with MP at 0.3 bpp (b) and decoded with scaling factors of $\sqrt{\frac{1}{2}}$ (a) and $\sqrt{2}$ (c).	160
7.25	Rate-distortion characteristics for MP encoding of the 256x256 <i>Barbara</i> and <i>Camera-man</i> images at 0.17 bpp, and truncation/decoding at different (smaller) bit rates.	162
7.26	Histograms of the MP coefficients when represented in HSV coordinates.	163
7.27	PSNR comparison between JPEG 2000 and MP. The PSNR has been computed in the CIELAB color space.	165
7.28	Top row, MP of sail with coefficients quantized in HSV color space. Bottom row, JPEG2000 for the same bit-rate.	165
7.29	MP of sail for 50, 150 and 500 coefficients.	166
7.30	Example of spatial scalability with MP streams.	166
A.1	(a) The “nice” functions $f_0(t)$ and $f_1(t)$ corresponding to the ℓ^0 and the ℓ^1 norm (b) An “exotic” sparsity measure, based on a mix of the ℓ^0 and ℓ^1 measures plotted in (a).	176
B.1	Anisotropic Refined atom.	178
B.2	Framework to calculate $\sigma_{s_{\max}}$	180
B.3	Gaussian atom.	181
B.4	Gaussian design frequency scheme. The bigger circle corresponds to the spectrum of the Gaussian with $\sigma_{s_{\min}}$ (and $\sigma_{f_{\max}}$ inversely) defined by the lowest frequency of the AR Atoms. The small circle corresponds to the $\sigma_{s_{\min}}$ (and $\sigma_{f_{\min}}$ accordingly) defined by the size of the image.	182
B.5	Incremental factor.	184
B.6	Gabor atom.	184

List of Tables

3.1	Comparison of the performance of different quadtree schemes in number of terms, distortion, rate and rate-distortion, for J quadtree decomposition levels and, if it proceeds, L levels of refinement.	51
6.1	Comparison of the computational time and the image quality obtained with the Genetic Algorithm, using 39 individuals and 75 generations, and the Full Search algorithm.	95
7.1	Comparison of spatial adaptivity of the MP encoder and JPEG2000. PSNR values are compared to quality obtained without transcoding (w/o tr.). 128 DF is the 128x128 image obtained with the JPEG2000 wavelet filter.	162

List of Algorithms

2.1	Curvelet algorithm	28
7.1	Adaptive Quantization	135

Abbreviations and Symbols

Mathematical notations

$\mathbf{A}, \mathbf{B}, \dots$	Matrices are denoted by bold capital letters
$\mathbf{x}, \mathbf{w}, \dots$	Vectors are denoted by bold lower-case letters and are column vectors
$\langle \cdot, \cdot \rangle$	Inner product operator
$\ \cdot \ $	Norm operator
\triangleq	"By definition"
$\lceil x \rceil$	the smallest integer greater than x
$\lfloor x \rfloor$	the largest integer smaller than x
$ \mathbf{A} $	the determinant of a matrix \mathbf{A}
$ Z $	the cardinality of a set Z
$E\{\cdot\}$	expectation operator
\mathcal{D}	(Redundant) dictionary of normalized atoms
Φ	Synthesis matrix of the dictionary \mathcal{D}
\mathbf{g}_γ	Atom belonging to the dictionary \mathcal{D}
\mathbf{s}	Signal of dimension N , that needs to be approximated

Abbreviations

HVS	Human Visual System
MP	Matching Pursuit
OBM	Object Based Model
HAM	Harmonic Analysis Model
HM	Horizon Model
GA	Genetic Algorithm
GA-MP	Genetic Algorithm based Matching Pursuit
FS-MP	Full Serach Matching Pursuit

Chapter 1

Introduction

1.1 Motivation

In this multimedia society, images are more and more everywhere. Images are very complex signals, with specific characteristics. The efficient approximation of natural images is a very complex task, and needs to be analyzed carefully. Dealing directly with natural signals is not straightforward, due to their complexity. In order to simplify this task, these signals are often modeled as a sum of simpler pieces, easier to manipulate:

$$\mathbf{s} = \sum_{m=0}^{M-1} c_m \mathbf{g}_{\gamma_m}, \quad (1.1)$$

where \mathbf{s} is the signal to approximate with at most M basis functions (with M often smaller than N , the dimension of the signal). c_m is the weight of the function \mathbf{g}_{γ_m} in this approximation and the set of $\mathbf{g}_{\gamma_m|_{m=0\dots M-1}}$ are the simple pieces used to represent this signal.

Physics provides numerous examples of such decompositions, mainly for wave propagation. The first example of such a signal simplification was the use of Fourier series for heat propagation. However, Fourier decompositions provide infinite series, and in order to be able to practically deal with the data, these series have to be truncated. Thus, instead of a perfect representation, the manipulated data is an approximation of the real world data.

In many applications, an approximation of a signal such as the one presented in Eq. (1.1) is needed, in order to have less terms to work with. There are several ways to achieve this approximation. One way can be to project the signal onto a fixed subspace of the dictionary (the set of basis functions). Doing this may not be optimal, because one can lose important information. For example, imagine that the signal is a sum of waveforms of different frequencies, and that you have a very high coefficient at frequency ω_h , which represents the main component of this signal. The set of basis functions used to represent this signal is the Fourier basis. Imagine now that the fixed subspace

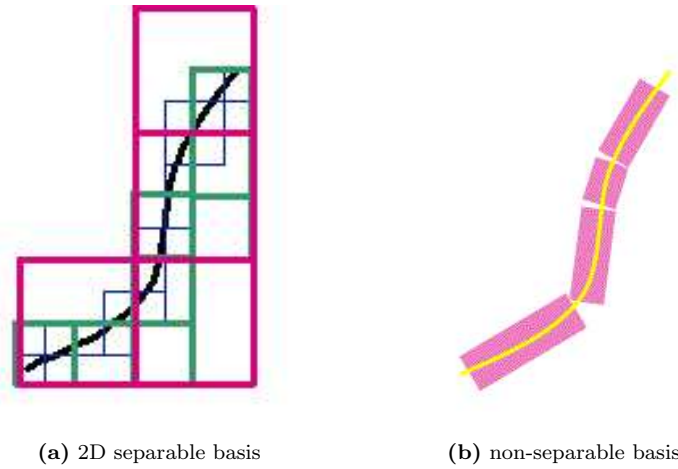


Figure 1.1: Wavelet versus edge adaptive scheme: Illustration of the fact that wavelets fail to detect the regularity of contours, while a set of basis functions that includes anisotropy and rotation would detect it.

where the signal is projected is the first M frequency components of the signal, $[\omega_0, \dots, \omega_{M-1}]$:

$$\hat{\mathbf{s}} = \sum_{m=0}^{M-1} c_m \mathbf{g}_{\omega_m} \quad \text{with} \quad c_m = \langle \mathbf{s}, \mathbf{g}_{\omega_m} \rangle, \quad (1.2)$$

where $\hat{\mathbf{s}}$ is the M term approximation of \mathbf{s} (with M smaller than N the dimension of the signal). If $\omega_h > \omega_{M-1}$, this component, which corresponds to the main component of the signal, is not detected. Thus, because of the fact of having a fixed sub-space choice, important information is lost in the simplified representation. This problem (the missing of important information in the final approximation) can be solved by performing an adaptive signal representation: instead of choosing the first M frequency components, the M basis functions that capture a maximum of energy from \mathbf{s} can be selected. If the signal is a pure sum of waves, the projection of the signal onto the Fourier basis with selection of the M functions with most energetic coefficients gives a good approximation of \mathbf{s} . But if the signal is no more a sum of infinite waves, but contains some spikes (for example, the signal is a piano concert, where the waves are created and disappear, and some spikes are present due to the impact of the hammer on the cords), the approximation of the M highest Fourier coefficients will no longer give a good signal approximation: all the transient information is lost.

Thus, being able to approximately represent a signal implies two main steps: the selection of the good set of basis functions to use in order to capture a maximum of energy with a small number of terms and the selection of the terms in this basis function set that better approximate the signal.

For 1D signals, wavelets are known to be able to sparsely represent signals that have both spikes and smooth zones (piecewise smooth signals) [28]. For 2D signals, however, wavelets lose part of their good approximation properties, due to the fact that they fail to detect the regularity of contours [16, 47, 51], as graphically shown in Figure 1.1 and stated in Chapter 3. The scope of this thesis is to efficiently represent images, which are 2D signals, that are often modeled as having smooth regions with smooth boundaries. As already said, standard separable basis fail to detect the smoothness of contours, and thus, do not provide sparse enough image approximations. Other schemes, such as frame methods, provide image approximations with a smaller number of terms than separable bases. Nevertheless, if the basis functions are not adaptive to edges, they

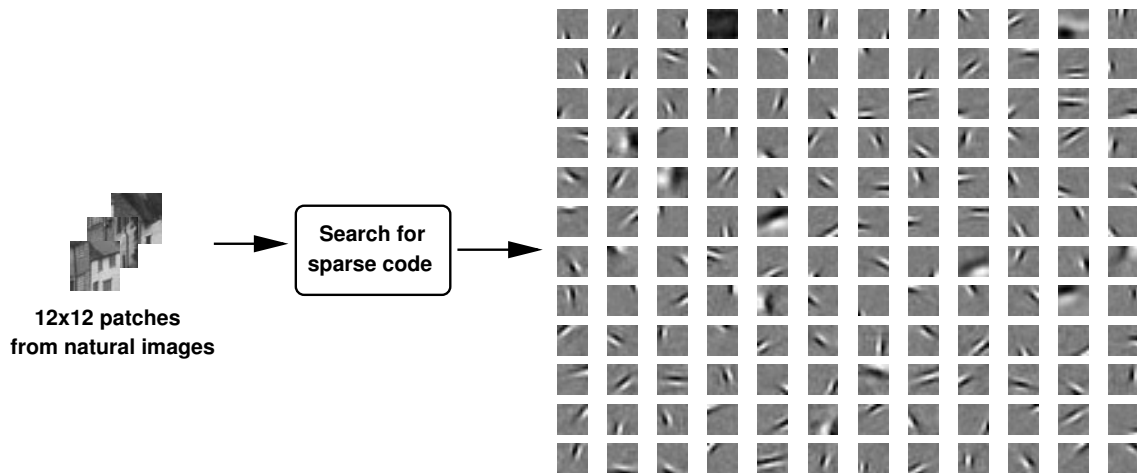


Figure 1.2: Basis functions obtained with Olshausen and Field’s method [139] from images of size 12×12 . The basis functions have been obtained by learning a sparse code from a set of natural images patches.

do not provide as good approximation scheme as adaptive edge approximations. Thus, the use of redundant sets of basis functions (called redundant dictionaries) formed by non-separable basis seems necessary, in order to give sparse image representations (see Chapter 3). This fact is evidenced in the observation of nature, and concretely, of the Human Visual System. Concretely, the learning of a set of basis functions that allows to sparsely represent natural images leads to anisotropic and rotated basis space-frequency functions, which are very similar to the visual cortex characteristic functions [138, 139]. Figure 1.2 shows the sparse basis functions learned from 12×12 natural images patches. It shows that anisotropy and directional structure of the basis functions appear, when one tries to learn the best set of basis functions to sparsely represent natural images. The main problem in the use of redundant dictionaries comes from a practical point of view: how can the image decompositions be computed, if frame methods cannot be used?

The fact of being able to sparsely represent an image is useful for many different image applications. Sparse image representations seem, of course, a useful tool for image coding at low bit-rates, as they provide a small error with a reduced number of basis functions. But low bit-rate image coding is not the only field that can profit from sparse flexible image representations or approximations. Sparse image representations are very useful for image denoising [50, 110, 178]. Being able to ensure that the set of basis functions used can capture the most important parts of the visual signal with a small number of coefficients allows to consider estimated coefficients smaller than a threshold as noise, and drop them without losing significant amounts of visual information.

Another desirable feature in order to have efficient image representation, which can be learnt both from the Human Visual System and the study of simplified image models, is the inclusion of geometry in the construction of the basis functions. This allows not only to have sparser image representations, but also to perform several interesting geometric operations directly in the transformed domain.

In addition, redundant dictionaries have proved to be useful in other applications, such as multimodal speaker detection [129] or video approximation [40].

The use of geometric based dictionaries leads to easier geometric image manipulations, which can help to construct super-resolution images or to implement new experimental segmentation techniques [126].

With the growing number of visual applications, especially over wireless networks, adaptive

low to medium bit-rate image coding becomes more and more necessary [149, 153]. In addition, as the number and the variability of end-user terminals grow, the image representation coding scheme should be able not only to efficiently approximate the image with few bits, but to be able to reconstruct images with different sizes and different rates, in order to adapt to the network conditions or to the end-user terminal conditions. An image coding scheme that allows for geometric transformations is a good solution to this new need.

Moreover, the amount of digital images that is produced and stored is growing in an exponential way. Storage capacity and network bandwidth are also impressively growing, thus, storage and transmission of the data is not the main problem nowadays. The main challenge nowadays is dealing and interacting with such huge amounts of data, which is growing and becoming more and more complex. In this situation, the use of smart coding and image representation techniques, that allow for fast browsing, recognition and indexing is required. Redundant dictionaries may provide a solution to this problem, by facilitating the inclusion of indexing or parametric information in the basis function themselves. This is, however, still an open problem.

1.2 Main contributions of the thesis

This thesis performs, in a first stage, a study about the features that basis functions should have in order to provide sparse representations of a piecewise-smooth image. This study emphasizes that, in order to have sparse image approximations, there is a need for edge adaptive basis functions, with geometrical structure. These basis functions need to have rotation, in order to capture local orientation of edges, and arbitrary anisotropy, in order to capture anisotropic image structures such as edges. Combining orientation and arbitrary anisotropy naturally leads to redundant dictionaries, as no known orthogonal basis can be created with all these features.

Signal decompositions over redundant dictionaries are not unique, and in most cases, just the sparsest solution (or a solution close enough to the optimal) is desired. In order to check in which cases a given image approximation in a redundant dictionary is the optimal one (the one that has the smallest residual for a given sparsity), a mathematical study about sparse approximations in the presence of noise has been performed. Putting together these conditions, an algorithm independent sparsity test is provided. This test is not only algorithm independent but it is also tighter than previous sparsity tests.

Once the basis functions properties needed to sparsely represent an image are understood, and a test that allows to check if the obtained solution is the sparsest, an algorithm to provide sparse image approximations has to be designed. Literature provides several algorithms capable, in some cases, of finding the sparsest approximation of a signal in a redundant dictionary. From all the existing algorithms, Matching Pursuit was chosen, due to the fact that it is an iterative algorithm (and thus, it is easier to deal with huge amounts of data) and that it has proved to be able, under certain conditions, to provide sparse signal approximations. Another main contribution of this thesis is the design of a dictionary that sparsely approximates images, and the Full Search Matching Pursuit algorithm that allows to find an image approximation in this dictionary.

Finally, this thesis provides a full flexible image coder. The core of this coder is the geometric image decomposition based on Matching Pursuit. A detailed study of the coding of the basis function indices is provided. This study has revealed that, to achieve good compression rates with the MP expansion over a geometric dictionary, there is a need for a good quantization of the Matching Pursuit coefficients. An enhanced *a posteriori* quantization scheme for MP expansions has been designed and implemented. This coding scheme is based on the theoretical study of the coefficient energy upper-bound and the empirical observations of the coefficient distribution and evolution

through the Matching Pursuit iteration number. A flexible image coder means an image coder that allows to perform operations in the transformed domain that are not straight-forward for every image approximation technique.

To summarize, the main contributions of this thesis can be concentrated in four main points:

- The use of redundant and parametric dictionaries for image approximation is proposed, with a study the properties of such dictionaries.
- The sparse approximation problem is studied, both at the theoretical and at the algorithmic level. At the theoretical level, the conditions under which the sparsest signal approximation is obtained are stated. At the algorithmic level, a MP algorithm for creating image approximations and its multichannel version are implemented.
- The design of a flexible image coder based on the previous findings is presented. This image coder is suitable for low to medium bit-rates, and it gives both rate scalability and the possibility of performing on the fly geometrical transformations in the coded domain.
- The extension of the previous coder for color images, profiting from the redundancy among different color channels.

1.3 Thesis outline

This thesis is organized as follows:

Chapter 2 presents three main topics that bring to a better understanding of the nature of images: the Human Visual System image representation at its different layers, different models that have been created in order to have efficient image representation, and different techniques that try to capture the nature of images. This chapter shows that the Human Visual System (HVS) has a complex image representation, adapted to the main needs of this system in order to process the information. It emphasizes that HVS tries to perform a sparse image representation, in order to have less data to process and be able to quickly analyze the environment. This same chapter presents then several image models that have been or are being used by the scientific community in order to be able to design appropriate image representations. It states that, from an image representation point of view, the main assumption that is made on images is that they have smooth contours and that they can be represented as a finite sum of basis functions. Finally, this chapter presents several image representation or approximation techniques that try to represent images as a finite sum of basis functions that capture the regularity of contours.

Chapter 3 performs a theoretical study in order to understand the basis functions properties that allow to represent an image with a minimum number of terms. This chapter shows that the fact of adding anisotropy and rotations in the basis functions support helps improving the approximation properties of the basis functions in use. A quantitative measure of the improvement in sparsity that anisotropic and rotated basis functions bring is given. The same chapter also studies whether these anisotropic and rotated basis functions have good compression properties, compared to the classical quadtree structures. It shows that adding anisotropy and rotation does not only improve the sparsity of the approximation but, if a smart coding scheme is used, can also give better Rate-Distortion behaviors. Of course, the fact of using anisotropic (with an arbitrary anisotropy rate) and rotated basis functions means that redundant dictionaries have to be used. The use of redundant dictionaries requires specific algorithms, able to solve this complex problem. **Chapter 4** explores the techniques that exist in order to recover the sparsest (or something close to the sparsest) approximation or representation of a given signal in a redundant dictionary. Several algorithms are

presented, together with their capability and their limitations, for recovering the sparsest signal approximation

Chapter 5 presents a new sparsity test in order to check whether the sparsest approximation for a given signal in a given dictionary has been recovered. This test, contrarily to existing ones, is independent on the algorithm that has been used in order to recover this approximation. It has been shown to be tighter than existing tests, but still, it becomes difficult to use for highly redundant dictionaries.

Chapter 6 introduces a sparse image approximation algorithm through a dictionary of anisotropic and rotated basis functions. This chapter introduces the dictionary of basis functions. A full search Matching Pursuit (MP) algorithm is presented in order to find the solution of the image approximation using such a dictionary. This full search algorithm uses specific dictionary features to speed up the search. The MP-based image approximation scheme is presented both for gray-scale and for color images.

Chapter 7 shows an image compression scheme that uses image decompositions obtained in Chapter 6. This chapter presents a full image coder that improves *a posteriori* quantization of Matching Pursuit coefficients by a novel enhanced reconstruction method. The best strategy for image compression through MP-based image approximations, together with the main problems that are found when performing image coding through this kind of approximations, are discussed in this chapter.

Finally, concluding remarks and a discussion about the possible future research directions to follow are given in **Chapter 8**.

Chapter 2

Image Models and Geometry

2.1 Introduction

This chapter is dedicated to a discussion of image models as the basis any image representation technique. The importance of geometry for being able to efficiently capturing important image informations is stated. This geometry is efficiently captured in the Humans Visual System (HVS). A brief description on how the HVS is given, and a discussion on which HVS features can be used as source of inspiration for designing an image representation model and technique is performed.

The chapter is structured as follows: First of all, a description of the HVS is given in Section 2.3. The HVS has been used for a long time as inspiration for modeling and representing images is given, thus, a brief review of how HVS analyzes images is useful. Section 2.4 gives a description of the most typical image model types. These models have been used to design several image representation techniques. Geometry in image representations will be discussed in Section 2.5, with an extension to consider non-geometrical image parts, such as textures. The Harmonic Analysis representation of an image will be presented in Section 2.6. Section 2.7 will present some image representation techniques that try to perform sparser image representations, by taking into account image features given by the models described in Section 2.4.

2.2 Motivation

An image is a very complex structure, often formed by a textured background with luminance changes and objects that overlap and occlude. Every object in the image is defined by its contours, shape, texture, smoother or sharper edges, etc. An image representation technique cannot take into account all the characteristics of images. These are much too complex objects to be able to optimize an image representation technique for all their different characteristics. Images need thus to be simplified through a model, which will be the basis for optimizing a given image processing operation (representation, coding, analysis, segmentation, etc.). The model will have to take into account the most relevant information of the image, while dropping its secondary features. A certain image characteristic will be considered secondary if it brings less information than other features, or

if the information that it provides is not relevant to the final goal or to the application. The model will have to be adapted to the application, and a careful choice of the most appropriate features has to be done.

When dealing with low bit-rate coding, for example, the first features that are necessary, normally, are the shape and mean luminance of objects, in order to be able to distinguish them. Shape is given by edges, so edge representation will be the primary goal. Textures become a secondary part of the image, because their contribution to recognition of an object is normally lower than what contours give. Textures, in addition, can be seen as a “messy” mix of edges, thus a technique that is optimal for edge representation will also be able to represent them, even though probably not in an optimal way. The image model that will be used in this thesis, therefore, will be mainly focused on edges, even though textures will not be completely forgotten.

2.3 How does the Human Visual System see Images?

In [124] a very simple, but also really appropriate, definition of vision is given: “Vision is the *process* of discovering from images what is present in the world, and where it is”. This is the *user* point of view reference. However, when dealing with signal processing, vision can be described as a process as well [123]: “Vision is a process that produces from images of the external world a description that is useful to the viewer and not cluttered with irrelevant information”. Describing vision as a process somehow defines *what* is done and *why*, and what has to be known, in order to find out the constraints and the rules that drive vision. In addition, it has a deeper consequence. In order to process something, there are two main things that have to be done: a *representation of the data* and an *algorithm*. There is normally a wide choice of representations for a given processing, and the choice of the processing algorithm will strongly depend on the particular representation that is employed. On the other hand, the choice of a given representation will make easier or more difficult the algorithm implementation for a given application. Of course, even for a given fixed representation, there will be several possible algorithms for carrying out the same process.

The HVS, which is the system in charge of vision, is a very complicated system. It has to perform several steps before achieving vision: first a data representation, which represents the data in an appropriate way in order to be able to correctly process it; and then a data processing, that will give the final information about the position, shape, color and even the identity or class of the objects or the scene it is observing. The HVS can be described with a block diagram with a first block that comprises the receptors, which perform a first processing of the data to transmit it to the brain (representation part), and a second block that performs the processing of the basic information (which may eventually drop useless information) and a third block for the interpretation of the information. Of course, this is the most basic scheme of the HVS, every block has several sub-blocks inside. Figure 2.1 shows a basic schema of the HVS. To achieve vision, the HVS must represent and process the received information. The representation has to be done in such a way that it helps the processing, and the processing will as well help the representation process.

The early stages of the HVS are very interesting for Image Processing, because in these first stages the representation of the data in a format that is useful for the further processing is computed. Some processing is also done in these stages. Several questions about the representation and the processing of the information arise. Some of these questions can be also very useful for Image Processing, in order to better understand the properties of images and how they can be represented in an efficient and flexible way. Some of these question are:

1. Has the interpretation of the information any influence on the processing of it?

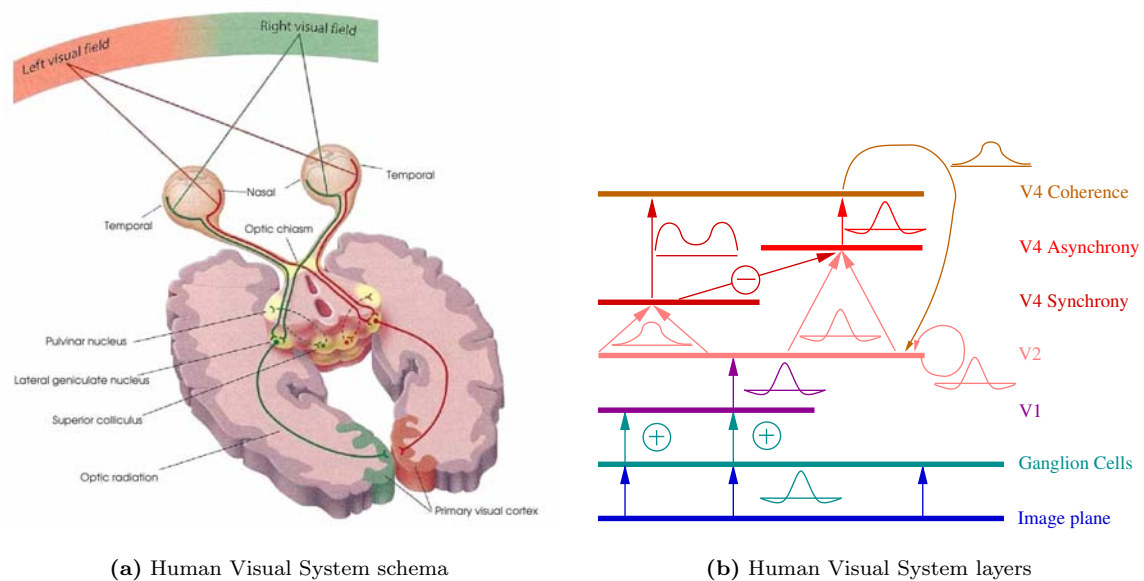


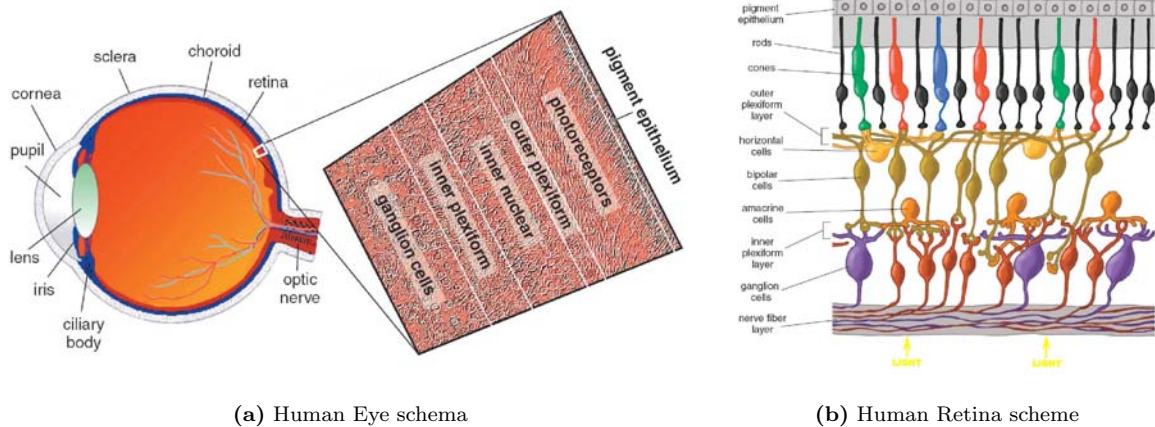
Figure 2.1: Human Visual System structure (a) shows the path the information follows, from the reception to the brain, in the HVS (Figure taken from [146]). (b) gives a schematic vision of the different layers of the HVS used for still image reception and processing, and a simplified version of the information transmission and processing.

2. And on the reception?
3. If there is a feed back, how much information is given in this feed back?
4. Which are the reception functions?
5. Are the reception functions optimal for image representation?
6. After the reception, the data has to be transmitted to the brain through the optical nerve. Is any coding of this data performed to send it through the optical nerve?

The answer to all these questions is not straightforward. Some of the HVS characteristics are very well studied and very well known, other are still under study. A very interesting source of information about the last advances on understanding the HVS can be found in [103]. In this section, a brief description of the different steps of information representation and processing in the HVS will be presented, with a special focus on the primary stages. Finally, some conclusions on what can be learnt from it, and how this information can be used, will be given.

2.3.1 Signal reception

The reception of the signal takes place in the retina of the eye. Figure 2.2 shows the path light follows until it reaches the receptors in the retina. Light goes through the pupil, then through the lens and hits the retina, where there are the receptors. The pupil controls the amount of light that hits the retina by narrowing or widening, and the lens focuses the light so that the image that hits the retina is clear. On retina lie the receptors of the light, that transform it in electrical signals which are sent through the optical nerve to the brain (see Figure 2.1).



(a) Human Eye schema

(b) Human Retina scheme

Figure 2.2: Information reception in the Human Visual System. Images are received in the Human eye (a), by projecting the light receiving trough the pupil to the retina, where there are different kinds of receptors (b).(Images taken from [102])

There are two kind of receptors in the retina, the cones and the rods. Rods are photoreceptors that contain the visual pigment, and are sensitive to blue-green light, with a peak sensitivity around 500 nm. As rods are only of one kind, they do not see colors, but they are highly sensitive photoreceptors and used for night vision (that is why in poor light conditions, we lose the capability of seeing colors). Cones are less sensitive, and they are used in daylight conditions. There are three kinds of cones: the long-wave cone (red light receptor), the middle-wave (green light) and the short wave (blue light). Cones are the basis of color perception in our visual system. Astonishingly, receptors are not in the front part of the retina, but in the back: light has to cross all the retina in order to reach the receptors (see Figure 2.2(b)). This is due to some physiological reasons, that are outside the scope of this dissertation.

Another difference among rods and cones, apart from the sensitivity, is the capability of detecting changes. Rods usually respond to relatively slow changes, while cones can detect rapid light fluctuations. The receptive field of both rods and cones is very narrow (they see just the spot of light that reaches them). The image is thus, for its reception, sampled, even though neighboring samples overlap.

The distribution of cones and rods in the retina is not homogeneous. Cones are mainly concentrated in the center part of the fovea (central part of the retina), while rods are more peripheral. This is the reason why peripheral vision does not allow us to clearly distinguish the color of an object.

Because of the presence of the optical nerve, there is a region in the retina that has no receptors. As the ganglion cells are in the front part of the retina, they have to cross the receptors layer in order to reach the brain. The point where all the nerves concentrate and cross the receptors layer has, thus, a hole without receptors. This causes a blind spot (an area that the human eye cannot see). This blind spot is normally compensated by eye movement, even though there exist some visual experiments that show how the blind spot has an effect over the observed image (see Figure 2.3, taken from [169]). This blind spot cannot be exploited for visual friendly image approximation, due to the fact that it will depend on the observer's distance, and that, as already stated, eye movement normally cancels this effect.

Once retina receptors have perceived a signal, they transmit this information to retina ganglion



Figure 2.3: The blind spot. This little experiment shows empirically the existence of the blind spot. Close your left eye and stare at the cross mark in the diagram with your right eye. Off to the right you should be able to see the spot. Don't LOOK at it; just notice that it is there off to the right (if its not, move farther away). Now slowly move toward the page. Keep looking at the cross mark while you move. At a particular distance (probably around 30cm), the spot will disappear (it will reappear again if you move even closer). The spot disappears because it falls on the optic nerve head, the hole in the photoreceptor sheet. (Image and caption have been taken from [169].)

cells, that will, consequently, transmit it to the brain through the optical nerve.

2.3.2 Information coding and transmission in the retina

Now that the reception of the visual information has been described, the question is how this information is processed in order to be sent to the Lateral Geniculate Nucleus, that will in its turn send it to the V1 cells, in the brain's primary visual cortex.

The human retina is a very complex mechanism, and it is not so easy to figure out how the information is processed before sending it to the brain. If the receptors are RGB, plus a sort of luminance (the rods), the way this information is sent is not so evident. Figure 2.2(b) is a schematic representation of the human retina with the different cells and the connections among them. Rods and cones (light receptors) transform light stimuli they can detect in electrical stimuli, and transmit it to the ganglion cells, that in their turn transmit all the information to the brain through the optical nerve. The way from the receptors to the ganglion cells goes through the bipolar cells. The horizontal cells and the amacrine cells also play a role in the stimuli processing before their transmission to the brain.

If the retina did not process the received visual information before sending it to the brain, the image would probably be blurred. It is already in the retina that some processing is performed in order for us to be able to define precise edges in images and focus on fine details. This processing is

not yet fully understood, but some parts of it seem to be clear. The first processing actually starts on the horizontal cells of the retina. While the receptive field ("area in which stimulation leads to response of a particular sensory neuron" [113].) of rods and cones is very limited, horizontal cells receive stimuli from several neighboring cones, and their receptive field is large. Single bipolar cells receive light from a handful of cones, and have a medium-size receptive field.

Single bipolar cells (which have an OFF or ON light response) would carry a fairly blurry response to their ganglion cells if horizontal cells did not add an opponent signal that is spatially constrictive, giving the bipolar cell a *center surround* organization. This means that the bipolar center signal is either ON or OFF, while the horizontal cells add a surround that is exactly the contrary (OFF or ON). For the retina cells, ON means detecting light areas on dark backgrounds, and OFF, the contrary.

From horizontal cells, the information is passed to ganglion cells. There are two types of ganglion cells in the human retina: ON-center and OFF-center. ON-center ganglion cells are activated when a spot of light falls in the center of their receptive fields, whereas OFF-center fire in response to light falling on their peripheral field. Ganglion cells have receptive fields shaped as a Mexican-hat. Ganglion cells in the fovea have much narrower receptive fields, due to the fact that they carry information from a single cone, but they carry in addition a color message regarding the type of cone (red or green) it comes from (see Figure 2.4). The receptive fields of two neighboring ganglion cells usually overlap, probably to increase reliability of the reception. Figure 2.4(c) shows this overlap.

Blue cones do not transmit the information in the same way as red and green cones do, they transmit it through a special blue cone bipolar cell to a different type of ganglion cell, which can carry both a blue ON and a yellow OFF response.

Ganglion cells transmit all the information that they receive to the brain through the optical nerve. This information is transmitted under the form of spikes. The meaning of these spikes is not really known yet, being the retina a complex mechanism that has still a lot to be studied (such as a previously unknown ganglion cell type that appears to behave as a giant photoreceptor itself, without needing input from rods or cones). The optical nerve then sends these spikes to the Lateral Geniculate Nucleus, where further visual processing is performed.

2.3.3 The Lateral Geniculate Nucleus

Optic nerve fibers from the eyes terminate at two bodies in the thalamus (a structure in the middle of the brain [92]) called the Lateral Geniculate Nuclei (LGN). The left LGN receives information about the Right Visual field, and the right LGN about the left visual field. LGN cells impulsive response has the same shape as retina ganglion cells response. Right lateral geniculate nucleus sends projections of the information received from the retina ganglion cells through the optical nerve directly to the Primary Visual Cortex. LGN receives also a strong feedback from the Primary Visual Cortex. The function of the LGN in the HVS is still unknown, so it is difficult to know which format has the information that the V1 cells receive. It is supposed that the function of the Lateral Geniculate Nucleus is to somehow control the amount and kind of visual information that will pass to the Visual Cortex.

2.3.4 Visual Cortex V1

The Primary Visual Cortex, also known as V1, striate cortex or area 17, is one of the parts of the brain that has been better studied. V1 cells receive the signal from the Lateral Geniculate Nuclei. It is not yet clear which format this information has, due to the fact that there are still some gaps in the knowledge about how the retina codes the information before sending it to the optical nerve,

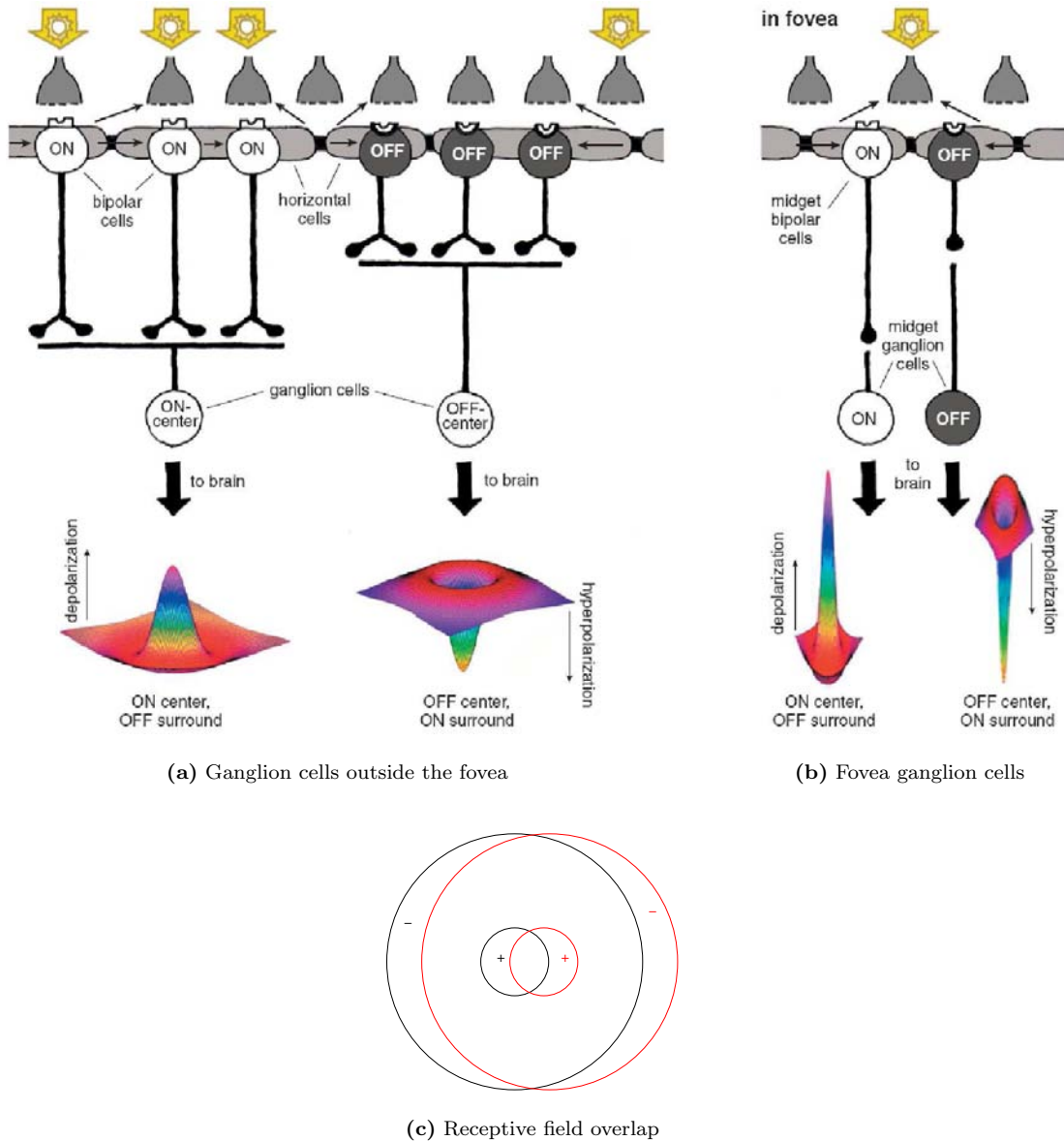


Figure 2.4: (a) Retina ganglion cells response, and its connections to bipolar cells for fovea and (b) for the rest non-foveal ganglion cells. (c) shows the overlap that can occur between neighboring ganglion cells, specially for those that are not in the fovea. (Images (a) and (b) taken from [102])

and there is almost nothing known about the function of the LGN. But one can know the response of V1 cells to a certain light stimulus, and numerous studies have given an answer to this question. It has been proved already since a long time now that the primary visual cortex cells are sensitive to bars of light of various orientations and sizes [96].

Primary visual cortex cells can be divided in two categories: simple cells and complex cells. Simple cells give a response when they see a bar that is oriented according to their angle. V1 cells receptive field seems to contain a central subregion that exerts an excitatory influence on the cell's response, and lateral subregions that exert an inhibitory influence. The most effective stimulus for

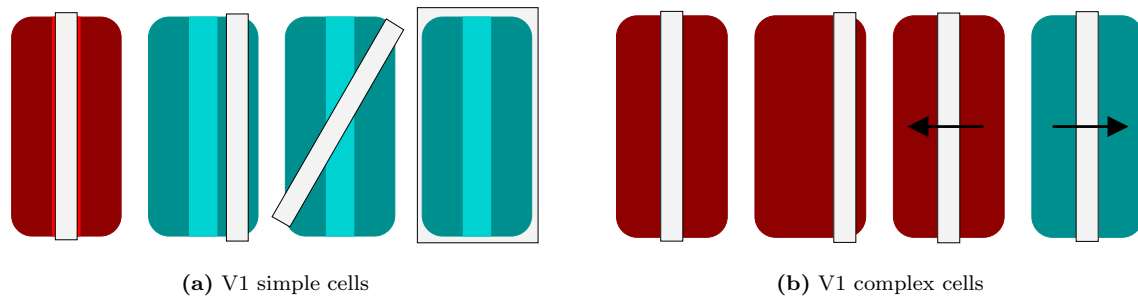


Figure 2.5: V1 cells response to several stimuli. Figure 2.5(a) shows the simple cells field and Figure 2.5(b) shows the complex cells field. The white stripe represents the stimulus received by the cell. The cell receptive field is represented in red when the stimulus excites the cell, and in blue when the cell does not respond to the stimulus.

this particular receptive field is one that puts a lot of light in the excitatory region, and only a little in the inhibitory region. The light must be bar-shaped, and have the right orientation, position and size in order to best excite these cells. Figure 2.5(a) gives an example of one stimulus that excites a simple cell, and of some stimuli that do not.

Complex cells are the most numerous in V1. Similarly to simple cells, they respond only to appropriately oriented stimuli, but the stimuli need not to be in a precise position. This kind of cells respond to any stimulus that falls inside the receptive field of the cell. Another difference of complex cells from simple cells is that the former normally respond to moving stimuli. Most of these complex cells respond only to stimuli moving in one direction, and do not give any response when the stimulus is moving in the opposite direction. Figure 2.5(b) shows several examples of stimuli and their effect on complex cells.

Simple cells respond to a stimulus that fall in a concrete part of its receptive field, independently on whether it is a moving stimulus or a static one, while complex cells respond mainly to moving stimuli. For static images, thus, is the simple cells response that is of interest. It is supposed that V1 simple cells perform a sparse coding of the received data [138, 139]. Applying the properties of this sparseness and trying to estimate which is the shape of the basis functions used gives interesting results for the basis functions shape (see Figure 2.6). As can be seen, these functions look quite similar to the theoretical response that was expected (with different orientations and sizes, and responding mainly to stimuli falling in the center of the impulsional response). This kind of basis functions will be thus studied for their use on sparse image representations.

It is supposed that the primary visual cortex has only a very basic color vision. Most of the responses of the V1 cells are color independent, however there are few cells in V1 have some basic color dependent responses.

The primary visual cortex is the part of the HVS that provides conscious vision. Removal of the primary visual cortex causes a loss of this conscious vision, but individuals that have not the primary visual cortex still react subconsciously to changes in light intensity, to movement in the visual scene, and even to some gross patterns of vision.

2.3.5 Visual Cortex V2

V2 is the next layer after V1. The information coded in V1 is somehow transmitted to V2. V2 is divided in four regions, that together provide a complete map of the visual world. Apparently,

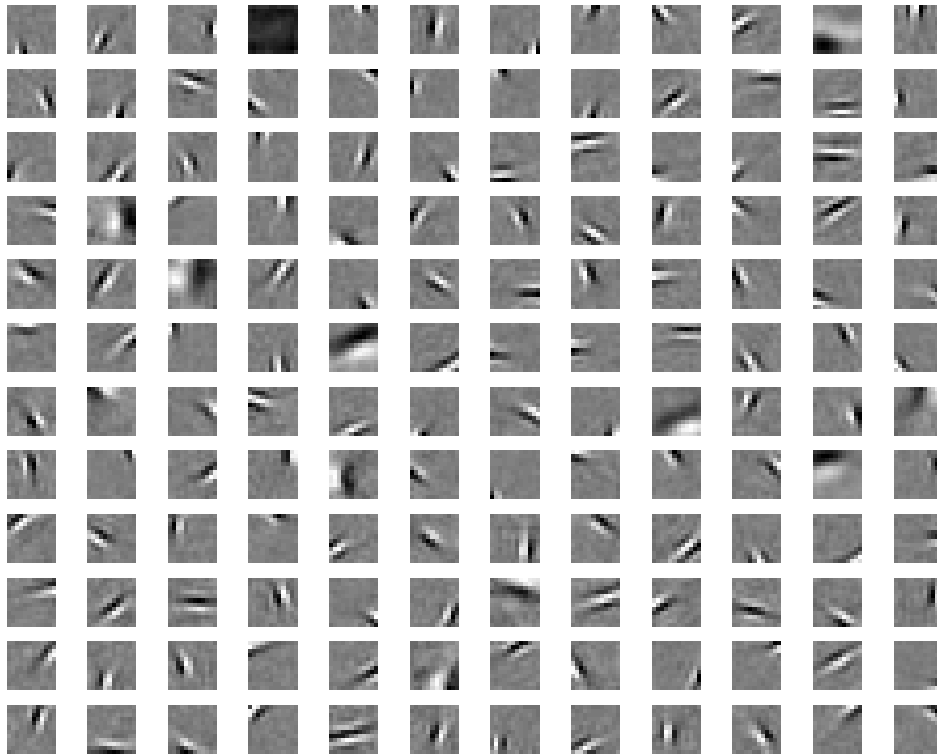


Figure 2.6: Basis functions obtained with Olshausen and Field's method [139] from images of size 12x12.

information mapping on V2 is the mirror image of the V1 map. Functionally, V2 has many properties in common with V1. Both zones are indeed tuned to orientation and spatial frequency. In addition, it seems that V2 is also tuned to color, in a more complex manner than V1 cells.

Recent research has shown that V2 cells are tuned not only to the orientation of contours, but also to the orientation of illusory contours [12]. It seems also that V2 cells are tuned for moderately complex patterns (textures), and, in contrast to V1 cells that respond only to one orientation, they may be excited by multiple orientations at different subregions within a single receptive field. However, there are still a lot of questions to answer about V2 cells.

2.3.6 V3 and above

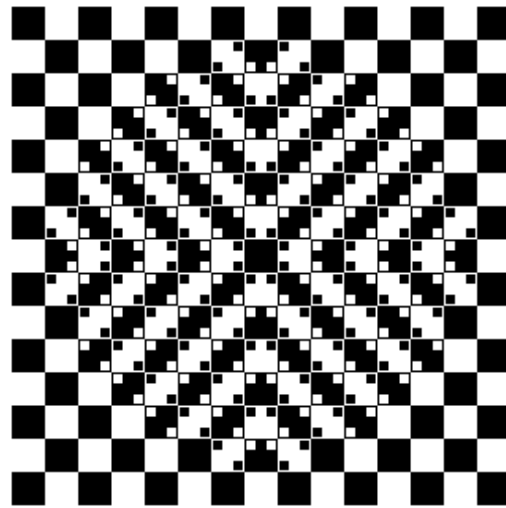
V3 cells transmission function is not yet well known. Most of the V3 cells are selective on orientation, and many of them seem to be tuned to motion and to depth. Few of them are sensitive to color. V3 cells receive stimuli from V2 cells, and send it to V4 cells, even though some interconnections of V3 cells with other visual areas exist.

V4 cells seem to have a role in color analysis (many of the cells are color selective). Cells in V4, though, have also complex spatial and orientation tuning, suggesting that the area is also important for spatial vision. The role V4 cells play in the process of vision is still under study.

The Middle Temporal area (also called V5) receives connections from V1 and from V2, and links these connections with the LGN. Most cells in MT (Middle Temporal) are tuned to motion, and the area can be divided into direction and axis of motion columns. It seems that MT is linked with the analysis of motion, but the fact that it receives connections from V2 (where few cells are sensitive



(a) How many faces do you see?



(b) Straight or wavy lines?

Figure 2.7: Some artistic images that profit from the HVS strong and weak points to create interesting visual effects. (a) takes profit from the ability of the HVS to detect faces to hide faces in the whole picture (some of them really evil). (b) exploits some of the lacks in information processing of the HVS to make the observer thing that the picture has wavy lines, when in fact all the lines are perfectly straight. Both figures are taken from [1], where there are many other examples of this kind of visual effects.

to motion) is somehow puzzling.

2.3.7 Recognition of objects and masking effects

Even though there are other layers in the HVS pathway, it is difficult to talk about the process in the brain, because it is still unknown. But even though the visual process is unknown, there are some psychovisual effects that are well known, thanks to some psychovisual experiments. Some of these effects will be presented here, for the sake of curiosity.

There are a lot of effects in the HVS that cannot happen before a certain recognition of the objects is performed. One clear example is the ability to recognize faces. In addition to the fact of recognizing the orientation, the motion and the color of a visual stimulus, it seems that the HVS has a sort of pattern recognition, that causes several effects, such as interpolation. There are many different images that profit from this ability of the HVS to create artistic hidden designs. An example of this is given in Figure 2.7(a).

The most usual masking effects of the HVS include these two facts: the surrounding color of an object affects the object perceived color and, at high frequencies, the contrast perception is weaker (contrast masking effect). There are other effects, such as the Weber law, which shows that the

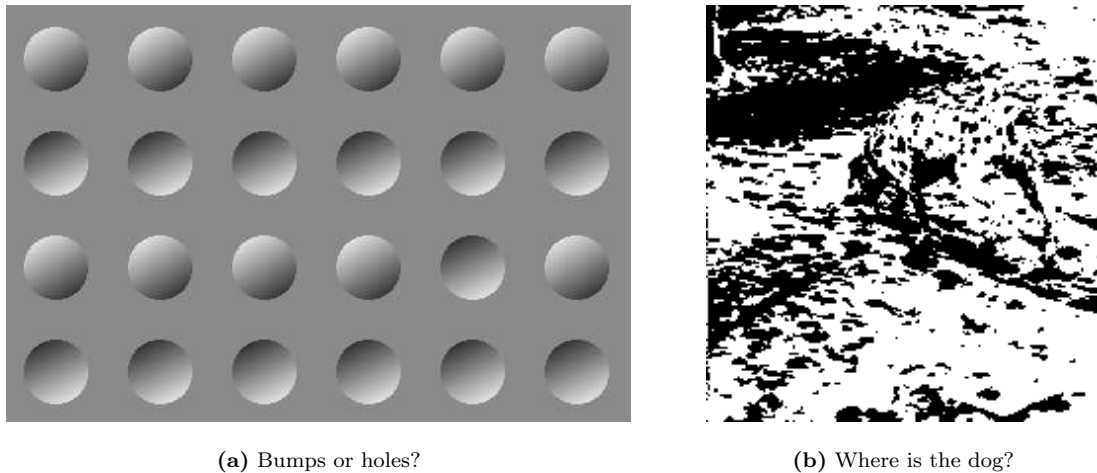


Figure 2.8: More optical illusions from [1]. (a) shows the effect shadows have on the perception of the third dimension. This figure is interpreted by the eye as two rows of bumps and two of dimples (generally, shadows on the bottom of the spot make the observer interpret it as a bump, and shadows on the top make it be seen as a dimple). But careful! The third row is not exactly what it seems... This shows that the Human Brain considers the world as being regular, and that an irregularity is only perceived when the scene is watched more attentively. (b) shows an example of the capability of interpolation and interpretation the HVS has. Even though the dog is not perfectly drawn, it is just a set of spots, a Human observer interprets the scene and perfectly sees the dog.

Human eye sensitivity to luminance is dependent on the surrounding luminance. There are three basic kinds of masking effects: spatial masking (a stimulus cannot be detected due to the presence of another stimulus nearby), temporal masking, which is not pertinent for still image, and pattern adaption (adjusting the sensitivity of the HVS in response to the prevalent stimulation patterns). All these maskings can be very useful for video coding, but for still image coding, as the observer can change its viewing distance and position, they are more difficult to use.

Other interesting visual behavior is the perception of depth and of the third dimension in some flat images. The perception of depth is strongly affected by the presence of shadows. The HVS usually associates a shadowed part with the lower part of an object (mainly due to the fact that the main natural source light, the sun, is always on the top of the objects). One example of this is given in Figure 2.8(a). Of course, there are many other examples of “lies” of the HVS. These false interpretations come basically from the evolution the HVS has followed. HVS has evolved so that it can interpret and detect any danger in the real world as fast as possible (survival of the fittest). When the situation differs from the most common natural environments, this evolution is cheating us and causing our brain to wrongly interpret some scenes. The interested reader can look at [11, 98] to read more about these psychovisual effects.

In spite of these errors, the HVS is capable of interpolating, segmenting and recognizing structures in images with an incredible ability. The typical example of face recognition (Figure 2.7(a)) clearly shows it. But the ability of the human eye to perceive objects that have contours ill-defined is really impressive (probably, if one could not perceive the silhouette of a predator in the middle of the grass died, so only humans that could interpolate, segment and recognized poorly defined objects survived). Figure 2.8(b) is a good example of this impressive capability.

2.3.8 Why psychovisual effects are so difficult to take into account?

As can be seen from the previous section, there are a lot of different psychovisual effects, from more or less sensitivity to colors and contrast according to the frequency or the surrounding color and pattern of the observed object. One could think that all these psychovisual effects could be used to improve image approximation techniques, so that the approximation error is not visible for the Human observer. There are indeed several studies that take profit from some of the features of the HVS information processing [131, 207, 208], but the truth is that most of the effects are too complex to be included in image processing techniques. There are several reasons for it: some of the effects depend on the Human observer, or on the distance between the observer and the image. Other effects are caused by complex interaction between different image components (such as the surrounding color plus a frequency component), and thus are very difficult to be detected by computers. In addition, image sampling complicates the detection of some visual effects, due to the loss of straightness in some contours.

It has to be noticed, too, that the fact of being able to take advantage of some of the HVS characteristics will strongly depend on the application. If the final objective of a given image is just to recognize the object that this image contains, there will probably be no problem on simplifying the aspect of the object. Natural images and pictures, though, are normally meant for different uses, depending on the final observer. One observer will have a detailed look at one object, while the other will just see the general aspect of a landscape. Because of this, it is difficult to use simplifications according to the surroundings or to the sensitivity to a given process, because depending on the final user distance, on whether the observer has a detailed look to one object or just a look to the whole image, the psychovisual effects that will appear will differ. In addition, some of the psychovisual effects are observer dependent (if the observer has trained the eye to look at a certain type of images, it will be easier for him to see a concrete detail). Psychovisual effects will be thus not taken into account here toward image simplification and approximation.

2.3.9 What can be imitated and to what point?

Resuming, the only parts of the HVS that are (partially) known are the retina and the V1 cells. Is this enough to get some inspiration for processing images? Well, it can be. It is clear that the HVS needs to represent the information such that it can easily be retrieved. To have an easy retrieval of information, there is the need of sending as few elements as possible. If sending too many elements, the brain might get lost into details, and not be able to process the important stimulus fast enough. So, it seems that a sparse representation of the images is needed, in order to be able to deduce the position, depth, and shape of the objects. As sparse decompositions are of interest for image processing purposes, and as the decomposition the HVS might be useful for detecting shapes and objects, it seems that an image representation system that mimics the first stages of the HVS can be very useful. Now, how can one imitate something that is not still well known? Even though it is true that there are a lot of gaps in the information, it is also true that there is a lot that is quite well known. So, we may not be able to exactly imitate the information processing that the HVS does, but it is also true that the raw information (in our case, the pixels of the images) is already different from the raw information retina receptors capture. As a consequence, what will be done, is not an exact imitation of the HVS information processing, but something that leads to an image representations similar to V1 cells output. Fortunately, this output is (quite) well known, even though some recent works show that less is known about V1 cells that it was expected [140].

To summarize, it would be desirable to have an image representation that is sensitive to edges, no matter what their orientation is, and that it is able to respond to different stimuli of different

size. This means that the representation should be able to capture the geometry of the edges. In addition, the representation should be sparse [138], in order to minimize the information to process or to send. Several techniques have tried to obtain representations that are sparse and that capture the regularity of edges. Some of these techniques will be explained in Section 2.7.

2.4 Image models

Every image representation technique is optimized according to a given model: Fourier decomposition [13] is optimized for invariant Harmonic signals, wavelets [28] for point-wise singularities, curvelets [16] are optimal for C^2 edges, edgelets [46] for piecewise linear discontinuities, etc. Every technique optimally represents signals that follow its “inspiration” model. The more accurate the model is with respect to the signal to represent, the better the technique will perform. The model, thus, becomes a very important part of the representation task. Ideally, one should have a simple model (something that is mathematically treatable), but as close as possible to a real image. A compromise has to be found, between an accurate model and an image model that is practical enough to be able to work with. In this section, a model of the image based on the local geometry and on Harmonic Analysis, and inspired by what is known about the HVS will be stated.

There are three typical ways of understanding an image: as a set of objects with certain interactions among them (Object Based Model), as a superposition of basis functions (Harmonic Analysis Model) and as a superposition of geometrical features (Geometrical Model). The three mentioned models are not exclusive. One of the three or a combination of them can be used, depending on the needs. For example, one could think of describing objects as a set of geometrical features or basis functions and then of using these objects to describe the scene. MPEG-4 [101], in its object based approach, is an example of this: it tries to combine object based representation and Harmonic Analysis [101, 142, 150].

The Object Based Model (OBM) is the way people often describe a scene. The Geometrical Model (GM) is often used in computer graphics, where one has the description of all the edges in a scene, their spatial position and orientation, so that occlusion can easily be described. This geometrical model is very useful for image synthesis, but it becomes difficult to use for representation purposes, because it needs an *a priori* knowledge about the contours, the object position, etc., that is normally not available in image representation.

When seeing an image as a set of objects that have certain interactions among them (OBM), a deep understanding of the scene is needed. This understanding implies an extremely complicated object segmentation and recognition. The OBM is the final model of the HVS, once all the psychophysical steps have been performed, and it implies a big storage of complex information. To illustrate it with a practical case, see Figure 2.9(a). Figure 2.9(a) represents an image with occlusions, image that can be very simply described by the objects that it contains. By saying that there is a brick wall holding a bike (placed in the center-left part of the picture), a tree growing behind the wall (placed behind the wall and just on the right of the rear wheel of the bike) and the sun slightly hidden behind the top-left part of the tree, the image is already quite precisely described. With this description, more or less everybody would reconstruct a similar image. This apparently simple description implies, though, a lot of very complicated steps. First of all, the image has to be segmented into objects that make sense (such as tree, wall, sun). These objects have been previously represented with some basis functions or some geometrical features. The simple object representation has then been compared to a “generic object type”, in order to be recognized and assigned to its concept (brown below and green above thing is a tree, for example). The objects need to be recognized even if they suffer from occlusions (such as the sun, that should be a perfect

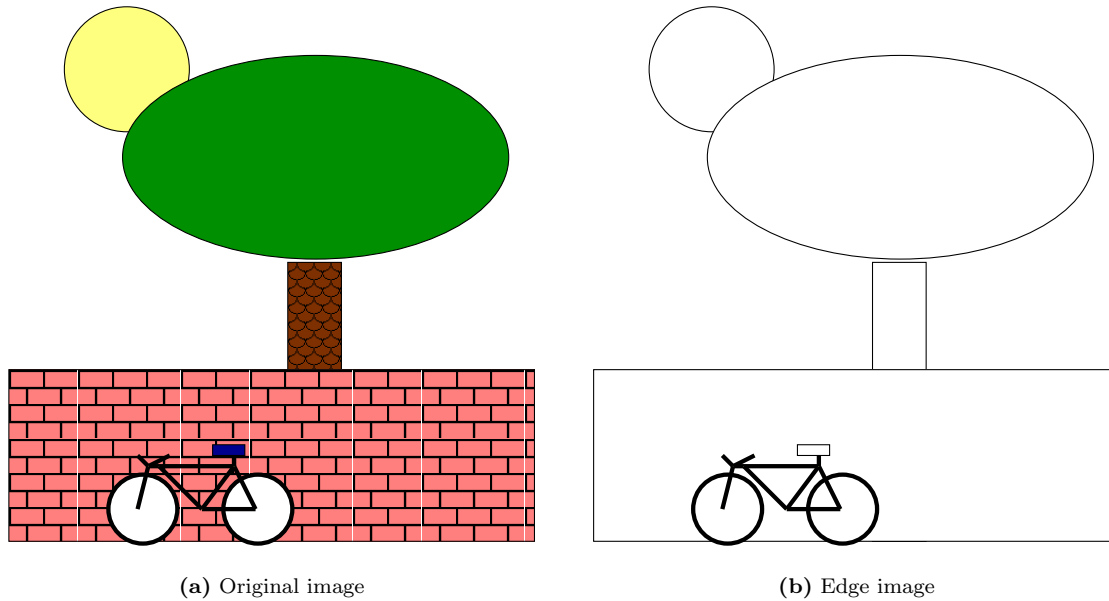


Figure 2.9: Example of image with occlusions. The bicycle is occluding a part of the brick wall which is occluding the tree, which in its turn, is occluding the sun.

circle, but, as it is slightly hidden behind the tree, it has a different shape). This implies a lot of *a priori* knowledge (the concept of brick, wall, tree, sun, etc.), which is very difficult to have in a simple machine (or even in a very sophisticated computer). In order to be able to perform this object recognition, first of all a huge database of “eigenobjects” has to be stored. This database has then to be browsed, and the good object has to be chosen. As the simple example presented here shows, the Object-based approach is not practical for image representation, and will not be of further discussion in this dissertation.

As recognizing and describing the objects of a scene is a too complex task, the image model that will be used in this thesis to represent an image will be more related to the basis layer of the HVS, that processes visual information before object recognition. These first steps of the HVS (HVS) are supposed to represent signals as a superposition of terms, using certain basis functions [123], which is in fact, the Harmonic Analysis approach. Geometry, though, is not completely forgotten, as the basis functions respond to certain geometric stimuli.

The Geometrical Model considers an image as a pure superposition of geometric shapes. This can be very useful for synthetic images, but it is not so straightforward to do with natural images. A bird, for example, has already a lot of geometric shapes in it (oval head, triangular beak, parallelepiped legs, etc.). Computer graphics often use this geometrical image representation, since the position where the objects and the contours will be placed is known. For natural image representation, though, it is not easy to represent images in this purely geometrical form, because the contours of the objects are often difficult to define exactly, and natural world is not as regular as we would like. In addition, as images are sampled versions of a continuous world, it is sometimes difficult to detect that a given set of points forms in fact a straight line (see Figure 2.10(a)). In order to have an efficient image representation, there is the need of a representation algorithm that is capable of optimally representing contours of objects (where most of the information of images is), independently from their size, position, angle or sharpness. A second step in this representation

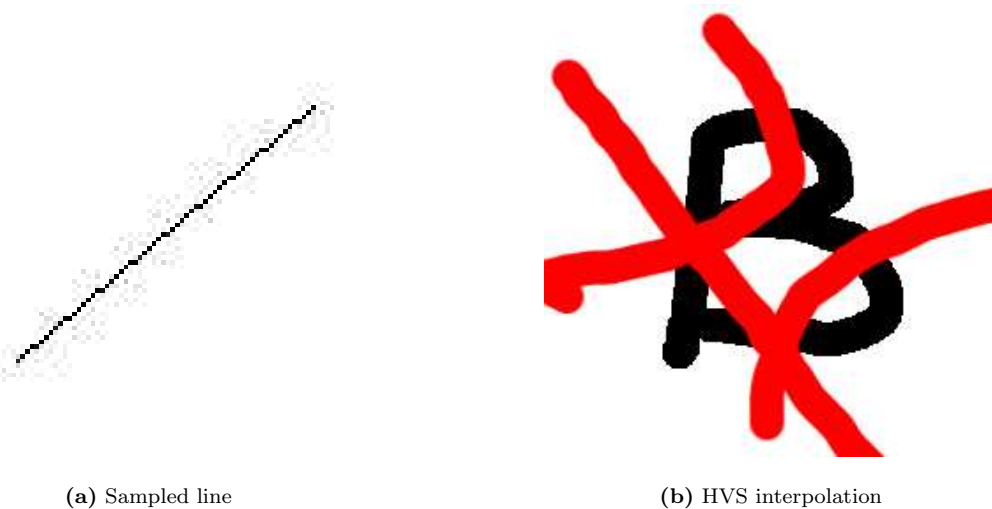


Figure 2.10: HVS versus machine perception of contours, lines and objects. (a) Example of the sampling of a straight line. The human eye easily perceives a straight line, even if it has some discontinuities. (b) Example of interpolation performed by the HVS. In spite of the occlusions, our brain is capable of seeing the B (thus, of interpolating edges where discontinuities are present).

process is to add textures and colors.

2.5 Introducing geometry in image representation

Considering geometrical features in image representation techniques is normally done by considering edge position, length, angle and, in some cases, curvature (thus, by considering just local geometry). The basis functions will not be able to recognize and represent geometrical forms, but just the position, length and orientation of edges, which define these geometrical forms. An image then is fully described as composed by two basic parts: the edges and the rest (which will include smooth parts and textures).

The separation of textures and edges in an image is not strict, and, depending on the scale, different edge-texture images can be observed from the same figure. For example, Figure 2.9(b) shows an edge representation of Figure 2.9(a), but one could think of putting the bricks (that in this edge representation have been considered as texture) as edges as well. It can thus be seen that even such an easy task of determining what are the edges of an image has no unique solution, and will depend on the scale and on the subjective opinion of the person that is doing it. An added complication to edge detection, when dealing with sampled digital images is that edges are not necessarily continuous, due to sampling (see Figure 2.10(a)). In addition, computers perceive images as a collection of pixels that have no relation to one another. It is easy for computers to detect a vertical or horizontal straight line (they follow a matrix line or column), but it is already more difficult to detect a diagonal line. Human eye is capable of detecting edges or straight lines in an image even if these are occluded or discontinuous (Figure 2.10(b) and [192]), but this operation can become very complex when dealing with rectangular sampled lattices. Other problems in detecting edges are due to the fact that they are not always sharp. Sometimes an edge is not a sharp change, but a smooth change that takes several pixels. Again, human eye easily recognizes the edge as the border of an object, but a computer can have problems at detecting it. Numerous techniques

exist, nowadays, for edge detection [10, 18, 114, 122, 125, 133], but it is preferable that an image representation technique does not depend on the ability of detecting edges.

Suppose now that edges are perfectly defined discontinuities, piecewise continuous. A geometrical representation of the edges has to be done. The easiest way to do this geometrical description is to consider edges as lines with a given angle, length and position. If there is a circle or an ellipse, as curvature becomes one more parameter to cope with, it will be split in small little edges which will be piecewise linear, and which will again have a given angle, position and length. Still, an edge model has to be defined. Typically, edges, for representation performance analysis are considered to be C^2 curves, or piecewise polynomial functions (or even piecewise linear) [42, 110, 155]. Some image and edge descriptions will be reviewed in this section, descriptions that can be used to perform a theoretical study of edge/image approximation properties of a given set of basis functions.

2.5.1 Horizon Model

The Horizon Model (HM) is the simplest image model. It considers images as purely edges. Edges, in this model, are defined by a continuous curve that separates two surfaces. The HM was first introduced as model of Boundary Fragments [107]. Since, it has been renamed *Horizon Model* [46] and used in numerous studies [42, 46, 64].

The HM supposes there is a function $H(x)$, called the horizon, defined on the interval $[0, 1]$. The image $s(x, y)$ is then of the form

$$s(x, y) = 1_{\{y \geq H(x)\}}, \quad 0 \leq x, y \leq 1. \quad (2.1)$$

This equation models a black and white image with an horizon, such that the image will be white above the horizon and black below. Several conditions can be imposed to the function $H(x)$, depending on the characteristics considered for the edge functions. Normally, the horizon is considered to be regular, and of finite length. Extra regularity can be imposed to the Horizon Function $H(x)$ by imposing it to hold Lipschitz regularity conditions [118] or simple derivability measures.

This HM is a very simple model, that does not match with real world images at all, because it handles only black and white images (Figure 2.11(a)). One could make the model more sophisticated by imposing different gray levels instead of black and white. The horizon would then separate smooth image regions (Figure 2.11(b)). An even more sophisticated model would be to replace the smooth regions by polynomially defined regions, that are separated by the Horizon function (Figure 2.11(c)).

Even though it cannot fully describe natural images, the HM is appropriate for studying the edge approximation properties of a given technique. This model is widely used for rate-distortion studies, or for studying the optimality of a given set of basis functions for edge representation. This HM will be used as well in Chapter 3 for studying the approximation properties of anisotropic and rotated basis functions.

2.5.2 Statistical model

There are two ways of modeling images statistically: considering images as a random variable [173, 177], or as a sum of basis functions which have certain statistical relation among them [203]. Considering an image as an observation of a random variable is useful for studying statistical image features, but it is difficult to perform an image representation directly with this model. The features obtained by the study of images as a random variable can be, nevertheless, used to improve other image representation techniques, by taking profit of the higher frequency of apparition of certain classes of events.

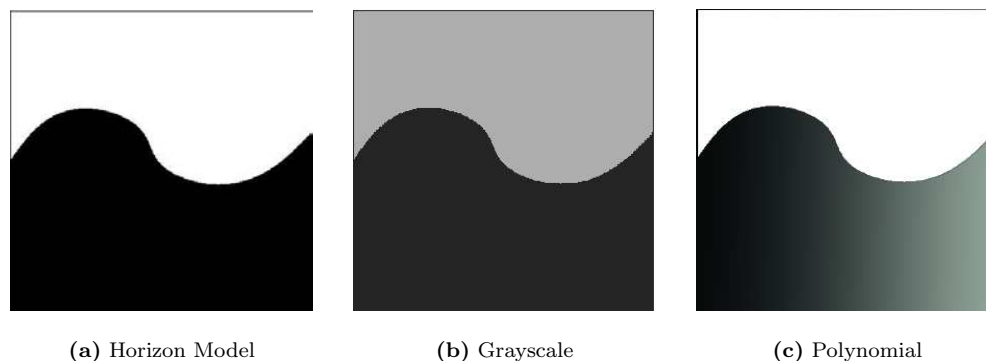


Figure 2.11: The Horizon model and some of its variations.

Considering images as random variables which have a certain probability distribution function is a usual way of modeling images [93, 173]. This model is supported by the experiments that show that HVS response may be adapted to efficiently represent signals with natural image statistics [168, 174, 175, 201, 202]. It is clear that there are events in images that are more likely to happen than others. Studying the event apparition allows, not only to be able to define the space of probable signals in the natural image set of functions, but to understand other factors, such as how the visual system of different species has been adapted to the needs and kind of signals that it receives. Considering the images as a pure random signal gives very interesting new ways of understanding some events. Representation techniques, though, cannot represent the signals directly as random variables, but can profit from the knowledge of the probability distribution of the events to code these more efficiently, using other techniques such as Harmonic Analysis for their representation. Some examples of methods profiting from natural image statistics for improving Harmonic Image representation are given in [14, 23, 154, 203], where the probability distribution of contours is used to code more efficiently the wavelet decomposition result. In [23, 154], geometrical events in images are modeled in the wavelet domain through Hidden Markov trees.

2.5.3 Other models

There are many other models of contours and of images. Probably the most popular one is to consider an image as a finite (or infinite) sum of functions, the so-called Harmonic Analysis model. This model will be presented in Section 2.6, where an image is considered as a sum of functions belonging to a dictionary. A dictionary is defined as a set of basis functions that at least expand the whole space [118]. An example of dictionary is an orthonormal basis. Other examples are a union of bases, or any set of functions that form a frame. This model is often combined with some geometrical models in order to study the optimality of the basis functions in use, or to find the characteristics the basis functions must have in order to efficiently represent the signal.

There are several other models for images, such as the fractal one, *e.g.* [22, 148, 163], which considers images as fractal curves. Apart from [148], that claims it can achieve lossless image compression, most of the fractal methods do not pretend to achieve lossless image representation, but a sparse image approximation instead. Fractals methods have been combined with wavelets as well [193].

2.5.4 Some remarks

As can be seen, there are several approaches for modeling an image. It has to be noticed, however, that one approach does not exclude the others. Indeed, for efficient lossless image compression, it is needed to combine the statistical model and the Harmonic Analysis model, so that any statical properties of the decomposition results are cleverly used for reducing the rate budget. In non-linear image approximation schemes, taking into consideration image statistics in the decomposition domain becomes less evident, even though these statistics can be cleverly used to design the set of basis functions to decompose the signal, for example. There is no model that can solve all the problems, one has to be able to properly choose the most appropriate image model and representation for a given application.

2.6 Representation through Harmonic Analysis

2.6.1 From Fourier to Wavelets

Mathematically, it is easier to deal with a sum of terms than with object description and object occlusions. Occlusions of objects could be dealt with logical operations, but the tools that exist for working with it are much less developed than the algebra and the function analysis used for standard representations. Harmonic Analysis considers images (and signals in general) as a finite (or infinite, in some cases) sum of basis functions \mathbf{g}_{γ_m}

$$\mathbf{s} = \sum_{m=0}^{M-1} c_m \mathbf{g}_{\gamma_m}, \quad (2.2)$$

where \mathbf{s} , vector of dimension N , is a discrete signal and \mathbf{g}_{γ_m} , $m = 0 \dots M-1$ are the basis functions. Often every term weight c_n depends on the scalar product of its associated basis function with the signal to represent. In some cases the weight is not exactly the scalar product (for example, for a frame it is the scalar product of the signal with the dual basis function instead of the signal with the basis function [52]).

Examples of such a representation are the Fourier decomposition [13], the Cosine Transform [3] or the wavelet decomposition [118]. In the discrete domain, all of them represent images as a finite sum of terms. This kind of decompositions are straightforward to obtain when working with an orthonormal basis. In such a case, \mathbf{s} will be represented as the sum of M elements (being $M = N$ the dimension of the space the signal is living in). In case M needs to be smaller than N (to have a sparse approximation of the signal), when dealing with an orthonormal basis, the only thing that has to be done is to keep the first M biggest coefficients in order to obtain the best approximation with this number of terms (it will be the set of coefficients that will provide the smallest decomposition error). Things get more tricky when dealing with non-orthonormal bases. This case will be explored in more detail later.

After Fourier Transform and all its relatives (Local Fourier Transform [118], Cosine Transform [118], etc.), wavelets were introduced [28]. Wavelets are also a representation of the image with a finite sum of terms, but wavelet basis functions give information about scale and position (and some information about the orientation as well, but quite limited for bi-orthogonal critically sampled wavelets [118]) instead of only giving information about frequency as for the Fourier Transform and its relatives (or frequency and position, when dealing with Local Fourier Transform).

Wavelets work optimally in mono-dimensional case (1D) [28]. However, when extending them to higher dimensions through tensorial product, they lose their optimality, due to the fact that they do not see the regularity of contours [20]. A simple example of this is given in Figure 2.12(a), where

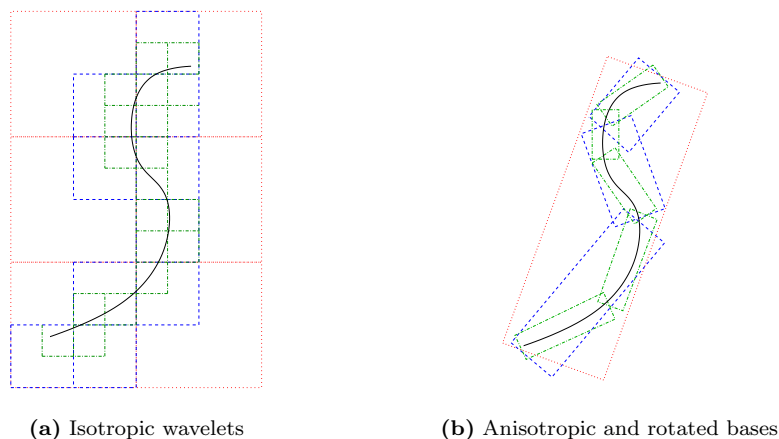


Figure 2.12: Edge decomposition with different kinds of basis functions. This example shows that when using Isotropic basis functions (a), the number of functions needed to represent a curve with a certain error is higher than when using anisotropic and rotated basis functions (b).

it can be seen that the fact of using isotropic basis functions causes a big concentration of functions along the edge. If the functions had been anisotropic and with rotations, the number of terms needed to approximate the edge with a certain error would have been much smaller (see Figure 2.12(b)). Several techniques have been studied in order to detect and exploit the regularity of contours. A brief overview of some of these techniques can be found in Section 2.7. Some of the techniques are partly inspired by the HVS, explained in Section 2.3

Existing tensor product based image representation tools are not yet optimal, because they fail to detect the smoothness of contours [16, 46, 64, 197]. In this scope, new non-linear techniques are being investigated, which would permit to optimally describe and represent natural and artificial images. Existing orthogonal separable bases are well adapted to mono-dimensional signal coding, but when dealing with higher dimensional signals, these bases fail to detect the smoothness of contours. With non-linear non-separable bases, this problem could be solved. This brought the interest for several other image representations, such as the redundant representations. In this thesis, the use of redundant representations with geometrically adapted functions for sparse image approximation will be studied.

2.6.2 Redundant representations

Redundant representations do not deal any more with bases, but with arbitrary dictionaries of functions. A dictionary of functions \mathcal{D} is a set of functions that at least expand the whole space. For example, an orthogonal base is a dictionary, but the union of two orthogonal bases is a dictionary as well.

The use of dictionaries is basically meant for overcomplete sets of functions. A dictionary can be formed by any set of functions, and even different function families can live in the same dictionary. In such a case, normally we speak about a dictionary formed by different subdictionaries. The dictionary will have to be adapted to the application. If what is needed is the possibility to sparsely represent an image, geometrical properties can be added to the basis functions. On the contrary, if what is needed is the security that the sparsest solution (the solution with less non-zero terms) has been found, it may be more interesting to use dictionaries with a limited redundancy [188].

For sparse image representation applications, the dictionary basis functions must have some very specific features. The HVS is a good source of inspiration for these basis function features.

The term redundant representation implies a signal representation on an overcomplete set of functions. This set of functions can be a frame [118] or simply a set of functions that expands the whole space .

2.7 Some geometrical oriented representation techniques

Several image representation techniques have been proposed that try to get inspiration from several features of the HVS in order to use geometrical information in images. The fact that edges are piecewise smooth functions is used in the design of the basis functions, in order to overcome the non-optimality of wavelets for signals of dimension bigger than one. Some of the most representative ones are discussed in this section. All the techniques presented here have something in common: they use a quadtree decomposition of the image (even curvelets, that look less similar to the quadtree structure, were first implemented through a quadtree-like method). This is done for computational reasons, but it is important to discuss whether the fact of using a quadtree causes these techniques to lose some approximation capabilities. The only technique that somehow manages to break the limitations of the quadtree structure is that proposed in [42], where an intelligent pruning of the quadtree partitions is performed. All the discussed techniques, however, allow to exploit the regularity of contours more efficiently than wavelets. Therefore, they have better approximation rates than wavelets with respect to the number of terms needed for the representation. Paradoxically, when applied to image coding, the performances of these techniques stay similar than the ones of wavelets, due to the great entropy coding of wavelet coefficients in JPEG2000 [2, 180].

2.7.1 Curvelets

While the localized Fourier Transform is a space-frequency representation and wavelets perform a scale-space representation of a signal, curvelets do a kind of phase-space representation. Curvelets were first presented in [16]. Later, [17] shows a more efficient curvelet computation, as well as some applications to edge-preserving image reconstruction.

A curvelet is nothing else than a localized ridgelet transform [15, 68], with an anisotropic a/a^2 pyramidal scheme. A ridgelet is a function that is smooth away of a discontinuity along a line. Ridgelets have infinite support, and similarly to the Fourier Transform, that has infinite spatial support and that has the windowed version, Curvelets are the spatially localized version of the ridgelet transform. The curvelet spatial localization is done in a very concrete way: by having an anisotropic support (outside this support the function will have zero value in the frequency domain), with anisotropy a/a^2 . This support is defined in the frequency domain, and can take several orientations.

Now, let $\Psi_{a,\theta,b}$ be a ridgelet [15]:

$$\Psi_{a,\theta,b}(x_1, x_2) = a^{-\frac{1}{2}} \psi \left(\frac{-x_1 \sin \theta + x_2 \cos \theta - b}{a} \right), \quad (2.3)$$

where ψ is a ridgelet function. And let define a ridgelet [15] coefficient as

$$\mathcal{R}s(a, \theta, b) = \int \mathcal{R}s(t, \theta) a^{-\frac{1}{2}} \psi \left(\frac{t-b}{a} \right) dt, \quad (2.4)$$

where $\mathcal{R}s$ is the Radon transform [32] defined by:

$$\mathcal{R}s(t, \theta) = \int s(x, y) \delta(-x \sin \theta + y \cos \theta - t) dx dy. \quad (2.5)$$

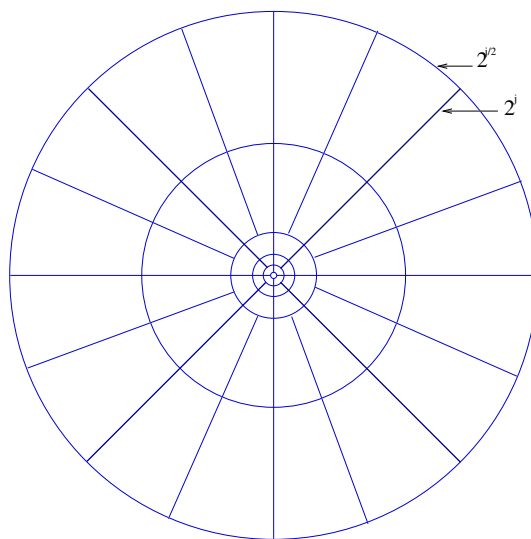


Figure 2.13: Curvelets: frequency domain division. Curvelets divide the frequency domain both in angle and in frequency rings. The number of divisions that increases with the frequency.

Analyzing the previous equations, one can see that the ridgelet transform is the application of a 1D wavelet transform to the slices of the Radon transform for every given angular variable θ . In \mathbb{R}^2 , ridgelets are constant along ridge lines $x_1 \cos \theta + x_2 \sin \theta = \text{constant}$ and they are wavelets with a scale a along the orthogonal direction. In the frequency domain, such ridgelets functions are localized in the corona $|\omega| \in [2^a, 2^{a+1}]$ and around the angle θ . In brief, performing the Ridgelet Transform in a finite support Fourier Window with dilation a/a^2 gives you have a curvelet.

In the frequency domain, a curvelet can be defined as:

$$\hat{\varphi}_{j,l,k}(\xi) = W_{j,0}(R_{\theta_{j,l}}\xi) e_{j,0,k}(R_{\theta_{j,l}}\xi), \quad (2.6)$$

where $W_{j,0}(R_{\theta_{j,l}}\xi)$ is the two dimensional window that divides the Fourier Plane in concentric frequency pseudo-cones with support $D_{j,l}$, and $e_{j,0,k}(R_{\theta_{j,l}}\xi)$ is a local Fourier basis on $L_2(D_{j,l})$. This is equivalent to filtering and then applying the windowed ridgelet transform to each band-pass image.

Curvelets are quite easy to implement in the discrete domain [178] through the *à trous* algorithm, as described in Algorithm 1.

Even though curvelets have a very good error decay with respect to the number of terms, they have not yet shown good compression properties, probably due to the fact that redundancy among adjacent coefficients has not been yet properly exploited.

2.7.2 Contourlets

Contourlets [43, 44] could be considered as digitally designed curvelets. Contourlets use, as curvelets, a parabolic scaling (the support has size a in the radial axis and a^2 in the perpendicular axis). Figure 2.14(a) shows the frequency domain division of contourlets.

Contourlets are attractive for image representation due to their good implementation features. First of all, they are defined via iterated filter banks, thus, they can be computed through fast algorithms, and profit from tree structures, similarly to wavelets. Another nice characteristic of contourlets is that they are defined on rectangular grids, thus all the phenomena that a rectangular

Algorithm 2.1: Curvelet algorithm	
1	apply the <i>à trous</i> algorithm with J scales
2	set $B_1 = B_{min}$
3	for $j = 1, \dots, J$ do
4	partition the subband w_j with a block size B_j and apply the digital ridgelet transform to each block
5	if $j \bmod 2 = 1$ then
6	$B_{j+1} = 2B_j$
7	else
8	$B_{j+1} = B_j$
9	end
10	end

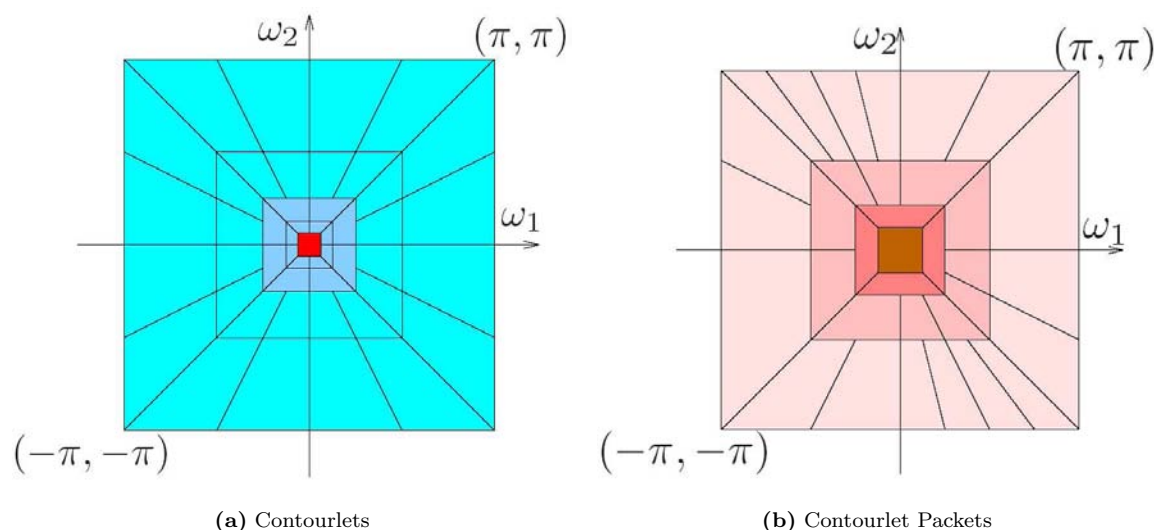


Figure 2.14: Contourlets and Contourlet Packets Frequency domain division. Contourlets perform a parabolic scaling of the frequency domain, while contourlet packets perform an adaptive division, dependent on the signal to code (image taken from [41]).

sampling introduces to images are already taken into account in the basis function characteristics. Contourlets have quite a good flexibility for image representation due to the fact that, for different directions, one can use different contourlet kernel functions (defined via filter banks). Finally, it is possible to implement compactly supported contourlet functions using FIR filters.

Contourlets have, similarly to wavelets, a tree structure. If this tree structure is used for coding purposes, the rate-distortion decay obtained is quite fast:

$$D(R) \approx (\log R)^3 R^{-2}. \quad (2.7)$$

Similarly to curvelets, contourlets impose the parabolic scaling. This may imply some losses of decay rate for a given type of signals. This is solved in Contourlet Packets [45], an adaptive scheme to select the “best” tree for directional decomposition which has been used for some authors to perform an image coding scheme [57]. Contourlet packets are formed by directional multiresolution

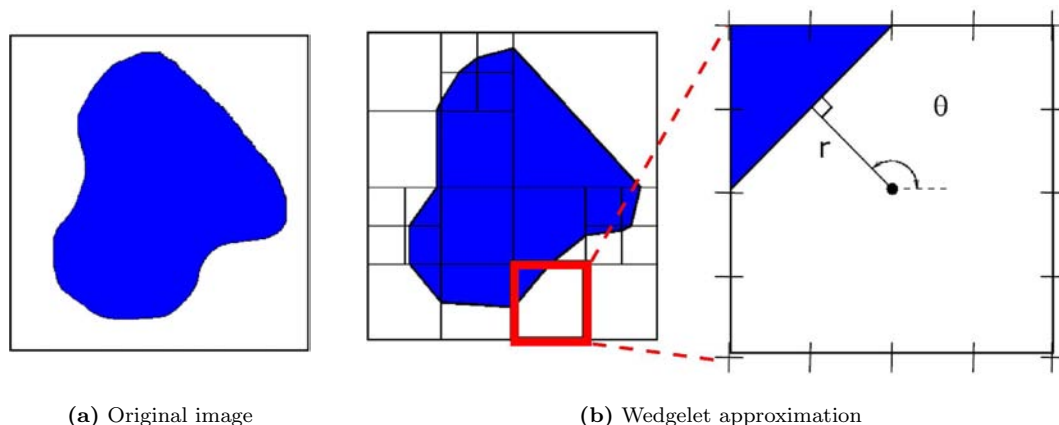


Figure 2.15: Wedgelet representation of a cartoon image. Figure 2.15(a) shows the original image and Figure 2.15(b) is its wedgelet approximation, with a zoom in one leaf to show a wedgelet description in the quadtree leaf (image taken from [6]).

elements with different shapes (aspect ratios) which do not necessarily satisfy this parabolic relation, as can be seen in Figure 2.14(b). Contourlet packets try to find a compromise between complexity of the decomposition and optimization in Rate-Distortion terms. Several variations of contourlets exist, such as the CRISP-contourlets (critically sampled contourlets) [115].

2.7.3 Wedgelets

Wedgelets take into account the regularity of contours by approximating them by piecewise linear segments [46]. A short description of wedgelets can be found at [33]. Wedgelets perform basically a quadtree of the image, but instead of taking as stopping criteria the fact that there is no contour inside the leaf, they take as stopping criteria that the leaf is uniform or that the representation error by approximating the contour in the leaf by a piecewise linear function is small enough. Figure 2.15 shows a wedgelet representation of a cartoon image.

The main characteristics of wedgelets is that they have a quadtree structure, thus, the deeper in the tree, the finer the curve approximation. To decompose an image with wedgelets, first a quadtree decomposition of the image is performed. Once this done, the wedgelet quadtree is pruned to approximate (adaptively) the local geometry. The leaves are then decorated with (r, θ) parameters, that will represent the angle of the wedgelet and the distance it has from the center of the image. This (r, θ) orientation is chosen from a finite dictionary, obtained by toroidal sampling.

Wedgelets are oriented towards image coding, thus all their division and pruning criteria are Rate-Distortion based. The first wedgelet pruning scheme considered, in its Rate-Distortion measure, leaves to be independent. So, the Rate-Distortion function of wedgelets was defined as:

$$D + \lambda R \approx \min_W \|f - \hat{f}_W\|_2^2 + \lambda[\#\text{leaves}]. \quad (2.8)$$

As every leaf is independent of the neighboring leaves, the total distortion can be computed as the addition of single leaves distortions:

$$\|f - \hat{f}_W\|_2^2 = \sum_{\text{leaves } i} \int_{W_i} (f - W_i)^2 = \sum_{\text{leaves } i} D_i. \quad (2.9)$$

This strategy only finds an optimal solution if the parameters are really independent among leaves, which is the case for distortion, but not for the rate. Generally, contours are smooth functions, and thus the angle of adjacent leaves does not vary drastically, and the variations among scales are also small. So, a better strategy would be to include in the rate-distortion equation the real rate needed to code the leaves, instead of the number of leaves. Of course, this increases the complexity of the coding, thus a compromise has to be found.

In spite of the fact that the cost of leaves is reduced if they are not coded independently and the correlation among adjacent leaves in a geometrical perspective is taken into account, for the computation of the cost this fact is not taken into account. The cost is also considered independent, so just the number of leaves will be taken into account in the Rate-Distortion optimization for performing the pruning of the quadtree, not the number of bits needed to code the leaves and the (r, θ) parameters of the wedgelet. This is done through a Multiscale Geometry Model, a coarse-to-fine prediction. This model will give higher probability to the model where the wedgelets draw continuous smooth curves, and very low probabilities to the model where the wedgelets draw chaotic structures. This is done through a joint wedgelet Markov Probability Model $P(W)$, and the new rate-distortion equation for wedgelets will thus be:

$$D + \lambda R = \min_W \|f - \hat{f}_W\|_2^2 + \lambda[-\log(P(W))]. \quad (2.10)$$

This gives better Rate-Distortion results than the first scheme, due to the fact that the probabilities better represent the Rate than the number of leaves. The wedgelet quadtree is again performed through dynamic programming, which gives a quite efficient computation.

To code images with edges and textures, Romberg, Wakin and Baraniuk [155] propose a dictionary formed by wavelets and wedgelets. Then wedgelets are used to code image parts which are formed basically by geometric objects, while image parts that consist in textures are coded with wavelets. In order to optimize the coding one cannot perform first the wedgelet approximation, subtract it to the original image and then perform a wavelet encoding of the residual, due to the fact that there are ridge artifacts that appear, and those are just as hard to code as edges. The strategy the authors follow then is the use of wedgeprints [206]. Wedgeprints are just the wavelet representation of a wedgelet, and can be followed through the different levels of a wavelet tree, following the same idea as footprints [51].

2.7.4 Footprints/Edgeprints

Footprints are groupings of wavelet coefficients along 1D discontinuities. The exact definition of a footprint for a piecewise smooth image is [51]:

Definition 2.1 *Given a piecewise constant signal s with only one discontinuity at position k , the footprint $f_k^{(0)}$ is the scale-space vector obtained by gathering together all the wavelet coefficients in the cone of influence of k and then imposing $\|f_k^{(0)}\| = 1$.*

In other words, for piecewise smooth images, a footprint is nothing else than the normalized set of non-zero (Haar) wavelet coefficients coming from a step function. Figure 2.16 shows a footprint in the time domain, and its non-zero wavelet coefficients.

Footprints can be designed for other kinds of signals than piecewise smooth functions. More interesting than piecewise smooth functions is piecewise polynomial functions. The idea is the same: take the wavelet coefficients that represent the discontinuity, group them, and use this group of wavelet coefficients as a basis function.

Footprints can be built from different kinds of wavelets: they can be built from a wavelet basis or from a wavelet frame [51]. The set of basis functions resulting from building footprints

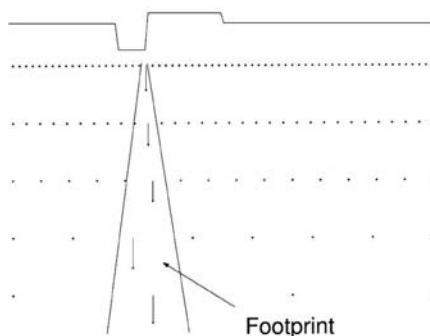


Figure 2.16: Footprint. Time domain (top) and wavelet domain (bottom) representation of the footprint $f_k^{(0)}$ with $N = 128$ and $k = 41$ (taken from [51]). Except for the case $J = \log_2 N$ (N being a power of 2), a footprint does not look like a pure step edge, since it does not include the scaling coefficients (image taken from [51]).

of piecewise polynomial signals is a redundant dictionary (an overcomplete set of basis functions), both if footprints have been built from a wavelet basis or from a wavelet frame. Footprints built from a wavelet frame, though, have the property of being shift invariant, while those built from a wavelet basis are not.

As footprints are not an orthonormal basis, but a redundant dictionary, an algorithm has to be found to be able to use them for signal representation. There are two proposed algorithms to expand a signal onto a footprint based dictionary: Matching Pursuit or Adaptive depth footprint pursuit [51].

Footprints are a good dictionary of basis functions for 1D signals, not for 2D. Edgeprints [50] extend footprints to 2D as follows:

1. Use footprints on each line and on each column of the image to detect singularities. Any singularity is characterized by $N + 1$ footprint coefficients.
2. Chain the footprint coefficients to form curves. The chaining of coefficients of the same order gives an edgeprint. Each curve generates $N + 1$ edgeprints.
3. Treat each edgeprint as a 1D signal and compress it using footprints
4. Compress the information related to the geometry of the curve using a sort of vertex-based shape coding [167].

Wedgeprints are inspired by footprints, but taking 2D signals instead of 1D ones. Wedgeprints are basically wedgelets that have been represented in the wavelet domain (see Section 2.7.3 and [155, 204, 205, 206]).

2.7.5 Bandelets

Bandelets try to exploit the fact that wavelets are optimal for 1D piecewise smooth function representation [109]. The basic idea of bandelets is that if wavelets can be aligned to the edge discontinuity, then, by putting the oscillatory part of the wavelet perpendicular to the edge, the minimum number of basis functions that have a value different from zero is minimized.

Bandelets basically align wavelets along discontinuities, similarly to contourlets [43, 44, 45]. Bandelets start by performing a quadtree partition of the image, but taking as homogeneity criteria

that in a given region, the gradient of the image has approximately a linear trajectory. This criterion can be slightly modified to have quadratic trajectories, or to have piecewise linear trajectories in a given region, but the idea remains the same. Once the partitions have been obtained, the intuitive idea of what bandelets do is to warp what is inside the region according to the gradient flow in order to have an horizontal or a vertical discontinuity only, and to code this discontinuity with the Warped Wavelet Transform [110], so that all the wavelets used have the oscillating part perpendicular to the discontinuity, and the non-oscillating part parallel to it. With this processing, the fact of using a lot of basis functions to refine a contour is avoided.

2.7.6 Some remarks

It is hard to say which technique performs better or which one is more adapted to the HVS, but there is one thing that has to be noticed. All the techniques use very similar approaches. Almost all of them keep using the quadtree decomposition, and once they have reached a certain level of division, they try to improve it by using the geometry in the small squares they have created. Bandelets do it with the gradient, wedgelets by approximating edges with piecewise-linear functions, etc. The only techniques that do not use the quadtree decomposition are curvelets and wedgelets, where the partitions are no more square, but with an anisotropy a/a^2 . Similarly to the other techniques, it uses regular divisions anyway.

Intuitively, a technique that does not use regular divisions, but that has a real anisotropy and rotation of the basis functions would have, for a low number of terms, a better approximation rate. In this thesis, this statement will be studied and verified.

2.8 Conclusions

Introducing geometry in image representation seems to be the best way to improve the representation performances, but it is not an easy task. The HVS, that is theoretically well adapted to process visual information, takes into account the geometrical features of images to represent and process the visual information.

This chapter has shown several image models, which try to take into account the main features of natural images. It has been seen that most of these models are naturally influenced by the way the Human Visual System perceives images. In the same chapter, several techniques that try to represent images in a more efficient way than standard separable bases have been introduced. It has been pointed out that one of the reasons why so many different image representation techniques appear is due to the fact that standard separable bases fail to sparsely represent natural images, due to the fact that they fail to detect the regularity of contours. Intuitively, one may achieve sparser image representations if, instead of using isotropic functions, anisotropic and rotated basis functions, able of taking profit of the regularity of contours, were used. Nevertheless, in order to fully understand if it is worth doing the effort of using a dictionary formed by anisotropic and rotated basis functions, a study of the gain that these new degrees of freedom bring has to be done. Indeed, using a non-orthogonal basis with anisotropic and rotated functions means that the decomposition of an image with these functions cannot be performed with standard techniques, and thus different algorithms are needed.

Towards geometric image representations

3.1 Introduction

This chapter performs a study of the gain of using anisotropic and rotated basis functions. Once this gain has been quantified, the Rate-Distortion (R-D) decay of this type of functions is studied as well, to see if a more efficient image coder can be implemented with this kind of basis functions. This is done through a very simplified image model: the Horizon Model (see Section 2.5.1). Textures are not taken into account for this study, which focuses on edges. Thanks to the horizon model and the simplification of the problem, it is possible to better understand the behavior of different types of basis functions in order to sparsely approximate an edge. In addition, some Rate-Distortion studies are presented, in order to check if the use of these dictionaries can give improved R-D results. This chapter does not pretend to give the final solution to image representation, but to perform another step towards the understanding of the behavior of edges in several image representation strategies. By directly introducing anisotropy and rotation in the basic quadtree structure, a toy model for an adaptive non-linear image representation tool is obtained. In order to better understand the improvement are due to anisotropy, and what improvements are given by rotation, two different schemes are presented: one that analyzes the effect of using anisotropic basis functions without rotation, and one that studies the use of anisotropic and rotated basis functions. Thanks to this, it can be seen that anisotropy makes the first derivative of the edge function appear in the R-D expression. Furthermore, when rotation is also included in the scheme, R-D is affected by the curvature of the edge, showing that the rate needed to represent a given contour is directly proportional to its geometrical complexity, which is coherent with the empirical and previous theoretical results. The different comparisons are done through different quadtree schemes: the classical quadtree [67], an optimized quadtree that includes refinement [42], both already in the literature, two new schemes that include anisotropy (the anisotropic quadtree and the anisotropic quadtree with refinement) and two new schemes that include anisotropy and rotation: the anisotropic quadtree with rotation and the anisotropic quadtree with rotation and refinement.

This chapter is structured as follows: Section 3.2 presents the details of the simplified image model in use for this study. With this model, the performance of every scheme is analyzed. Sections 3.3 to 3.8 analyze the performances of six different schemes: isotropic quadtree, optimized isotropic quadtree (isotropic quadtree with refinement), anisotropic quadtree, anisotropic quadtree with refinement, anisotropic quadtree with rotation and anisotropic quadtree with rotation and refinement. Section 3.9 gives some empirical rate-distortion results for the different quadtree schemes, showing that the theoretical bounds are coherent with what is observed in practice. Finally, Section 3.10 draws some conclusions.

3.2 Image model in use

In order to be able to theoretically study the behavior of basis functions with anisotropic and rotated support, and to compare it to other types of basis functions, a simplified image model is needed. Indeed, it is much too hard to theoretically estimate the number of terms needed to represent an image without performing a simplification of the problem. In this chapter, in order to mathematically study the behavior of the different types of basis functions, the image is modeled through the Horizon Model for piecewise smooth images (see Section 2.5.1).

Thus, the image $s(x, y)$ is considered to be defined on the interval $[0, 1]$ and to take the form:

$$s(x, y) = 1_{\{y \geq h(x)\}}, \quad 0 \leq x, y \leq 1, \quad (3.1)$$

where $h(x)$ is the Horizon function, and is defined in the interval $[0, 1]$ as well. In order to be able to perform all the needed computations, some assumptions are made for $h(x)$: $h(x) \in \mathbf{C}^p$ is p -times continuously differentiable and has finite length inside the unit square.

A way to represent this black and white image is through a quadtree decomposition, which is in fact a toy model for wavelets. [42, 172] already demonstrated that the R-D of the quadtree decomposition of this model decays as R^{-1} , and that the number of terms increases as 2^J , with J the number of decomposition levels. To improve these results, [42] proposed a refined quadtree scheme, introduced in Section 3.4 in order to compare its behavior with the anisotropic and rotated basis functions.

3.3 Quadtree

The standard quadtree is a very well known representation technique [42, 67, 172]. This model is presented here just because it is a toy model for wavelets. Just as reminder, the quadtree works as follows: Check whether a given partition contains an edge or a smooth region. If there is a smooth region, code the color of this smooth region (black or white) and stop dividing. If there is an edge inside, dyadically divide the x and the y axis of this partition, giving four smaller partitions. The same procedure is then performed with the smaller wished partitions. The quadtree stops when the smaller partition resolution has been achieved, or if only smooth parts are found (see Figure 3.1(a) for an example of quadtree decomposition).

This very simple algorithm is a toy model for wavelets: wavelets too, when they “see” an irregularity, keep on dividing (have non-zero coefficients at higher resolution), while they stop dividing when the surface is homogeneous enough (has more regularity than the number of zero moments of the wavelet) [28, 118].

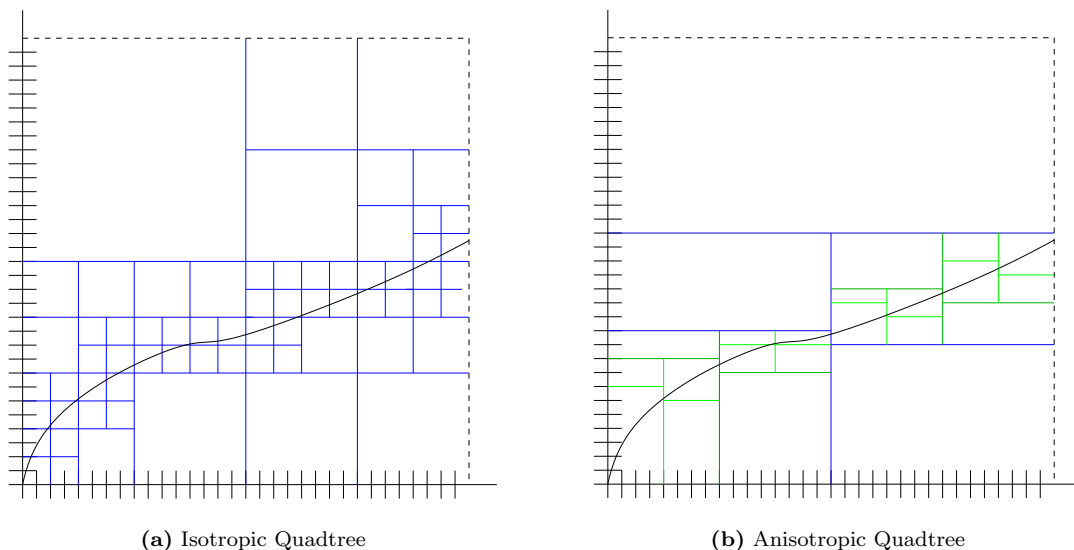


Figure 3.1: Isotropic and Anisotropic quadtree. (a) Example of the behavior of the isotropic quadtree, or standard quadtree, when trying to approximate an edge. (b) Example of the behavior of the anisotropic quadtree when trying to approximate an edge.

3.3.1 Number of terms in the decomposition

For the quadtree with refinement, the number of terms that are considered to play a role in the decomposition is approximately the same as for the quadtree without refinement. In fact, the number of non-zero coefficients is approximately the same, but the non-zero coefficients bring more information, and it will be seen that, for the same resolution, the final distortion is smaller in the refined quadtree than in the standard one. Of course, when a Rate-Distortion study is performed, the fact that the non-zero coefficients of this kind of quadtree bring more information implies that more bits are needed to code them. These additional bits are, anyway, useful, due to the significant decrease in distortion they bring.

In brief, for the optimal quadtree, one has to count $N_J \sim 2^J$ non-zero coefficients when J levels are used as well.

3.3.2 Distortion of the decomposition

The final maximum distortion of the isotropic quadtree corresponds to the area of the non-zero components at the last resolution. The area of a partition at the last resolution (and so the distortion that one coefficient gives) is $D_1 = 2^{-2J}$, if J levels of refinement are performed. The total distortion will correspond to the sum of the distortions of all the non-zero components. Thus, for the isotropic quadtree, the total distortion is:

$$D = D_1 \cdot N_e \sim 2^{-2J} \cdot 2^J \sim 2^{-J}. \quad (3.2)$$

3.3.3 Rate-Distortion behavior

The Rate-Distortion behavior of the isotropic quadtree is already well known [42, 160, 161]. The quadtree has a rate that increases with the quadtree depth as:

$$R = 2 \cdot N_a + J \cdot 2^J, \quad (3.3)$$

where N_a is the number of all nodes and $J \cdot 2^J$ is the bits needed for the edge regions. Edge nodes need J bits (for the corresponding J quantization levels) and all the nodes need two supplementary bits to indicate if a given node is a flat white region, a flat black region, an intermediate node (a node that must still be divided) or an edge node (a node that has reached the smallest resolution and that has the edge inside). The dominant term is finally the rate needed to code the edge nodes, and so, the rate as a function of the depth levels in the quadtree can be written as follows:

$$R \sim J \cdot 2^J. \quad (3.4)$$

Writing J as a function of the Rate and substituting it in the expression of the distortion (Eq. (3.2)), one gets a Rate-Distortion behavior of [42]:

$$D(R) \sim \frac{\log R}{R}. \quad (3.5)$$

Notice that the R-D behavior of the quadtree does not depend on the regularity of the edge.

3.4 Optimized quadtree

The Optimized Quadtree [42] decomposition is similar to the Quadtree algorithm, but the stopping criteria is not “there is an edge inside”, but “there is an edge inside that is not linear”. Once the edge inside the partition is a linear segment (or very close to a linear segment), it stops dividing, and codes the *refinement*: where the edge crosses the quadtree partition.

Recently, a more evolved version of this quadtree has been proposed [172]. This optimized quadtree, once the quadtree decomposition has been achieved, joins the consecutive partitions that are similar enough (that have an edge that crosses the neighbor partitions at the same point and with the same slope or that are flat and with the same color). This is another very interesting way of having anisotropic partitions, but will not be analyzed here, because it is not a toy model for a concrete image representation scheme.

3.4.1 Number of terms in the decomposition

For the quadtree with refinement, the number of terms that are considered to play a role in the decomposition is approximately the same as for the quadtree without refinement. In fact, the number of non-zero coefficients is approximately the same, but the non-zero coefficients bring more information, and it will be seen that, for the same resolution, the final distortion is smaller in the refined quadtree than in the standard one. Of course, when a Rate-Distortion study is performed, the fact that the non-zero coefficients of this kind of quadtree bring more information implies that more bits are needed to code them. These additional bits are, anyway, useful, due to the significant decrease in distortion they bring.

In brief, for the optimal quadtree, one has to count $N_J \sim 2^J$ non-zero coefficients when J levels are used as well.

3.4.2 Distortion

When using refinement, the distortion for a given number of terms is smaller, of course. If there are J quadtree levels, and L refinement levels (2^L refinement segments inside a quadtree partition), the distortion for one coefficient will no longer be $2^{-J} \times 2^{-J}$, but $2^{-J} \times (2^{-J} 2^{-L})$, so for one coefficient the distortion is $D_1 = 2^{-2J} 2^{-L}$. The final distortion when L levels of refinement and J quadtree depths are used is then [42] :

$$D = D_1 \cdot N_e \sim 2^{-2J} 2^{-L} \cdot 2^J = 2^{-J} 2^{-L} \quad (3.6)$$

In fact, adding this refinement is equivalent to introducing, in a limited way, some anisotropy and rotation. It can be seen that, as expected, adding refinement in the quadtree causes that, for the same number of coefficients, the distortion is smaller. [42] demonstrated that this reduction in distortion implies an improvement in R-D performance as well.

3.4.3 Rate-Distortion behavior

The rate estimation of the optimal quadtree based compression is performed similarly to the rate estimation of the normal quadtree, but taking into account the L refinement bits that are needed. The total rate of this kind of quadtree can be written as [42]:

$$R = 2 \cdot N_a + 2L \cdot 2^J \sim L \cdot 2^J. \quad (3.7)$$

Taking the expression of the distortion (Eq. (3.6)), and, without loss of generality, considering that $L = J$ to simplify it, one gets that the Rate-Distortion behavior of the optimal quadtree is:

$$D(R) \sim \frac{\log R}{R^2}, \quad (3.8)$$

which is an improvement with respect to the normal quadtree. Once more, the expression is independent on the regularity of the curve. This expression is approximately the characteristic wedgelet R-D [46, 204, 205], which use a very similar representation scheme.

3.5 Anisotropic quadtree

In the anisotropic quadtree [63, 64], the x axis keeps on being dyadically divided, similarly to the standard quadtree technique, but the y axis partition depends on the position of the edge. The size of the partition height is always chosen so that the superior and inferior horizontal limits of the partition are located as close as possible to the edge but without crossing it. Once a first partition has been built following this scheme, the dyadic rectangle containing the edge will be divided in two along the x axis. The partitions are refined and further divided, until the desired accuracy is reached (*i.e.* until either the height or the width of the rectangle has size 2^{-J} , with J the maximum number of refinements allowed). Figure 3.1(b) shows an example of an anisotropic quadtree decomposition of a contour.

3.5.1 Number of terms in the decomposition

As can be seen in Figure 3.1(b), the number of edge partitions that are found in the anisotropic quadtree is exactly the number of rectangles that have achieved the last resolution.

The iterative anisotropic quadtree algorithm is going to stop when the height or the width of the rectangle achieves the minimum resolution allowed. The width of the rectangle is always divided

dyadically, except in the case where the height of the rectangle has already achieved the minimum resolution. The height of the rectangle depends on the slope of the edge that is inside. If the edge inside the present partition is a straight line with slope one (a line with a 45° angle), the partition will be a perfect square. If the slope of the line is smaller than one, the partition will be wider than higher. On the contrary, if the slope is bigger than one, the partition will be higher than wider. With this, one can see that the slope of the curve that is being represented influences the height of the partition, and thus the number of divisions that participate in the representation of a curve. A curve that is almost flat is represented with few partitions, while a curve that is very steep needs more divisions to achieve the smaller resolution.

The slope of a curve is computed through the first derivative of the curve. To get an upper-bound of the number of coefficients that participate in a decomposition, one has to consider the worst case, and so, to take the maximum value of the first derivative of the curve that is being analyzed inside each partition. An even worse upper bound is the maximum of the derivative of the function $h(x)$ in the interval $[0, 1]$. The height of a partition will then be smaller or equal to its width times the maximum of the first derivative. So, for resolution j , the height of a partition satisfies:

$$\left\lceil \frac{2^{-j} \cdot |h'(x)|}{2^{-J}} \right\rceil \leq 2^{J-j} \lceil |h'(x)| \rceil. \quad (3.9)$$

Thus, the quadtree will perform at most

$$J_{\text{stop}} = J + \lceil \log_2 |h'_{\text{max}}(x)| \rceil \quad (3.10)$$

iterations. There are two cases to be considered: $h'(x) < 1$ and $h'(x) \geq 1$. If $h'(x) < 1$, the above stopping criterion is reached before the maximum quadtree depth allowed J . On the contrary, if $|h'_{\text{max}}(x)| > 1$, it means that $J + \lceil \log_2 |h'_{\text{max}}(x)| \rceil > J$, thus, the quadtree stops at J . With this, one can deduce that the number of edge partitions in the anisotropic quadtree is:

$$N_{e_J} \leq \min \left\{ 2^{J + \lceil \log_2 |h'_{\text{max}}(x)| \rceil}, 2^J \right\}. \quad (3.11)$$

Notice that the case $h'_{\text{max}}(x) = 0$ is very special, because it means that with only one iteration, the minimum resolution has been achieved. The above expression is then non valid, and simply $N_{e_J} = 1$.

3.5.2 Distortion

The distortion of the anisotropic quadtree is not very different than the distortion of the isotropic quadtree because, at the lowest resolution, the total area of the partitions that have an edge inside is almost the approximately same for both schemes. The fact of having possibly larger end partitions is compensated by the fact of having less partitions, at the last resolution. The distortion of one coefficient in the anisotropic quadtree without refinement can be upper-bounded by the area of the partition. Two cases have to be distinguished: $J_{\text{stop}} = J$ or $J_{\text{stop}} \leq J + \lceil \log_2 |h'_{\text{max}}(x)| \rceil$. For the first case, the distortion of one coefficient is:

$$D_1 \leq 2^{-J} \cdot 2^{-J + \lceil \log_2 |h'_{\text{max}}(x)| \rceil}, \quad (3.12)$$

where 2^{-J} is the width and $2^{-J + \lceil \log_2 |h'_{\text{max}}(x)| \rceil}$, the height. As the number of nonzero terms for this case is 2^J , the final distortion is:

$$D \leq 2^{-J + \lceil \log_2 |h'_{\text{max}}(x)| \rceil} \sim 2^{-J}. \quad (3.13)$$

For the second case, the distortion of one coefficient is also:

$$D_1 \leq 2^{-J} \cdot 2^{-J + \lceil \log_2 |h'_{\max}(x)| \rceil}, \quad (3.14)$$

but in this case, the width is $2^{-J - \lceil \log_2 |h'_{\max}(x)| \rceil}$ and the height is 2^{-J} . The number of nonzero coefficients for this second case is $2^{J + \lceil \log_2 |h'_{\max}(x)| \rceil}$, so the final distortion turns to be:

$$D \leq 2^{-J}. \quad (3.15)$$

With this, it can be seen that, in terms of sparsity, introducing anisotropy without rotation is not really worth: the reduction of the number of terms is not significant, and the distortion remains approximately the same. Thus, if anisotropy is introduced, it has to come together with rotation in the basis functions support.

3.5.3 Rate-Distortion

The number of bits needed to code an anisotropic quadtree can be expressed as:

$$R \leq \sum_{j=0}^{J_{\text{stop}}} 2^j \cdot 2 \cdot N_{\text{bits}_j}, \quad (3.16)$$

where N_{bits_j} counts the number of bits needed to code the height and the position of the partition, and equals

$$N_{\text{bits}_j} = J - j + \lceil \log_2 |h'(x)| \rceil. \quad (3.17)$$

The sum in Eq. (3.16) is a finite arithmetic sum, and its solution is:

$$R \leq 2^{J_{\text{stop}}} \cdot 2 [J + \lceil \log_2 |h'(x)| \rceil - J_{\text{stop}} + 1] - 4J - 4 \lceil \log_2 |h'(x)| \rceil - 4 \quad (3.18)$$

$$\sim 2^{J_{\text{stop}}} \cdot 2 [J - J_{\text{stop}} \lceil \log_2 |h'(x)| \rceil + 1]. \quad (3.19)$$

As it has been done to compute the distortion and the number of terms, two cases have to be distinguished: $J_{\text{stop}} = J$ and $J_{\text{stop}} = J + \lceil \log_2 |h'(x)| \rceil$. In both cases, though, the asymptotic behavior of the rate with the quadtree depth is:

$$R \sim 2^J. \quad (3.20)$$

In order to see the evolution of the distortion with the rate, it will be necessary to find $J(R)$, and to substitute it in the expression of the distortion. It is easy to see that $J \sim \log R$, and thus, the final distortion turns out to be:

$$D(R) = \begin{cases} 2^{\lceil \log_2 |h'(x)| \rceil} R^{-1} & j_{\text{stop}}=J \\ R^{-1} & j_{\text{stop}}=J+\lceil \log_2 |h'(x)| \rceil \end{cases} \quad (3.21)$$

3.6 Anisotropic quadtree with refinement

The anisotropic quadtree with refinement starts with the same procedure as the isotropic quadtree with refinement. Once the partitions have been generated, and the last resolution have been achieved, it codes the position of the edge at the border of the rectangles and approximates it as a straight line.

3.6.1 Number of terms

Similarly to the isotropic quadtree with refinement, the anisotropic quadtree with refinement uses the same number of terms as the anisotropic quadtree. The only thing that changes with respect to the anisotropic quadtree without refinement is the information that every term brings, and thus the final distortion.

3.6.2 Distortion

Again, the distortion of the anisotropic quadtree with refinement and the distortion of the isotropic quadtree with refinement are equivalent. For one coefficient, the distortion of the anisotropic quadtree with refinement is defined as:

$$D_1 = \int_{x_-}^{x_+} h(x) - r(x) dx, \quad (3.22)$$

being x_- the lower bound in the x axis of the partition under analysis, x_+ the upper bound, $h(x)$ the edge function and $r(x)$ the straight line used to approximate $h(x)$. The total distortion is then the sum of the distortion of all the partitions in the finest resolution. In the above integral, $r(x)$ takes the following expression:

$$r(x) = Q[h(x_-)] + \frac{Q[h(x_+)] - Q[h(x_-)]}{x_+ - x_-} (x - x_-), \quad (3.23)$$

where $Q[h(x)]$ is the quantization of $h(x)$ with L refinement levels and L quadtree depths, and can thus be written as $Q[h(x)] = h(x) \pm 2^{-J-L}$. Substituting in Eq. (3.22) $h(x)$ by its third order Taylor expansion in the interval middle point $x_c = \frac{x_+ + x_-}{2}$:

$$h(x) = h(x_c) + h'(x_c)(x - x_c) + \frac{h''(x_c)}{2}(x - x_c)^2 + O((x - x_c)^3) \quad (3.24)$$

and taking into account that $x_+ - x_- = 2^{-J_{\text{stop}}}$, one obtains the distortion for one coefficient:

$$D_1 = \frac{-h''(x_c)}{12} \cdot 2^{-3J_{\text{stop}}} \pm 2^{-J-L} \cdot 2^{-J_{\text{stop}}} \cdot 2 + O(2^{-4j}) \quad (3.25)$$

As the final distortion of the anisotropic quadtree with refinement is the sum of the distortions of all the partitions at the finest resolution, it becomes:

$$D \sim 2^{-J} 2^{-L} + \frac{h''(x) 2^{-2J_{\text{stop}}}}{12}, \quad (3.26)$$

where the quadtree is performed with a deep J and L levels of refinement inside the smallest size partitions. Taking $J = L$, one finds that the final distortion is:

$$D \sim 2^{-2J} \quad (3.27)$$

It can be seen that the distortion for the anisotropic quadtree with refinement is similar to the isotropic quadtree with refinement, and the number of coefficients is similar as well.

3.6.3 Rate-Distortion

Let J be the number of bits used for quantizing. If N_{a_j} is the number of all the rectangles at iteration j and N_e the number of edge-rectangles (rectangles that have an edge inside), the number of all nodes N_a and of edge nodes N_e is:

$$N_{a_j} \sim 2^j, \quad (3.28)$$

$$N_e \leq 2^{J+\lceil \log_2 h'_{max} \rceil}, \quad (3.29)$$

where J is the maximum number of decomposition levels allowed. As next iteration rectangle is inside the previous iteration one, the number of bits needed to code the size of each rectangle decreases with the iteration number as:

$$N_{\text{bits}} \leq J - (j - 1) + \lceil \log_2 h'_{max} \rceil. \quad (3.30)$$

Taking into account the bits needed to code the rectangle sort (black, white, edge or intermediate) and its size and position, the total bit-rate needed to code the anisotropic quadtree is given by:

$$R = (2 + N_{\text{bits}})N_a + 2L \cdot N_e, \quad (3.31)$$

where L is the number of bits used to code the position where the edge crosses the dyadic rectangle. The first term counts the bits needed to describe the partition position, and if 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 Eqs. (3.28), (3.29) and (3.30) with Eq. (3.31), one obtains:

$$R \leq \sum_{j=0}^{J+\lceil \log_2 h'_{max} \rceil} [(2 + J - (j - 1) + \lceil \log_2 h'_{max} \rceil)2^j] + 2L \cdot 2^{J+\lceil \log_2 h'_{max} \rceil}. \quad (3.32)$$

By developing the sums as arithmetic series, the previous equation turns to:

$$R \leq (2 + J + \lceil \log_2 h' \rceil)(2^{J+\lceil \log_2 h' \rceil+1} - 1) \quad (3.33)$$

$$- [2^{J+1+\lceil \log_2 h' \rceil}(J + \lceil \log_2 h' \rceil - 1) + 2] + 2^{J+1+\lceil \log_2 h' \rceil}L. \quad (3.34)$$

Simplifying, Eq. (3.33) gives:

$$R \leq 2(3 + L)2^{J+\lceil \log_2 h' \rceil} - J - \log_2 h' - 4, \quad (3.35)$$

and so:

$$R \leq 2(L + 3)2^J + \lceil \log_2 h' \rceil. \quad (3.36)$$

When considering $L=J$ and high bit-rate the previous equation turns to:

$$R \sim J \cdot 2^{J+\lceil \log_2 h' \rceil}. \quad (3.37)$$

This result is equivalent to the quadtree result, but with a term $2^{\lceil \log_2 h' \rceil}$ which comes from the fact of using irregular h axis division. Taking the distortion given by Eq. (3.27), and considering again that $L = J$, one obtains:

$$D(R, f) \leq C2^{-2J}. \quad (3.38)$$

And from the previous expression and Eq. (3.37):

$$D \sim \frac{2^{2\lceil \log_2 h' \rceil} \log_2^2 R}{R^2}. \quad (3.39)$$

In the above expression, it can be seen that the rate-distortion of the irregular quadtree is similar to the one of the regular quadtree but with a factor $2^{2\lceil \log_2 h' \rceil}$ that comes from the use of anisotropy. The use of rectangles when dividing gives already a factor $2^{\lceil \log_2 h' \rceil}$ and the coding of the position where the edge crosses the border of the rectangle gives the other term $2^{\lceil \log_2 h' \rceil}$.

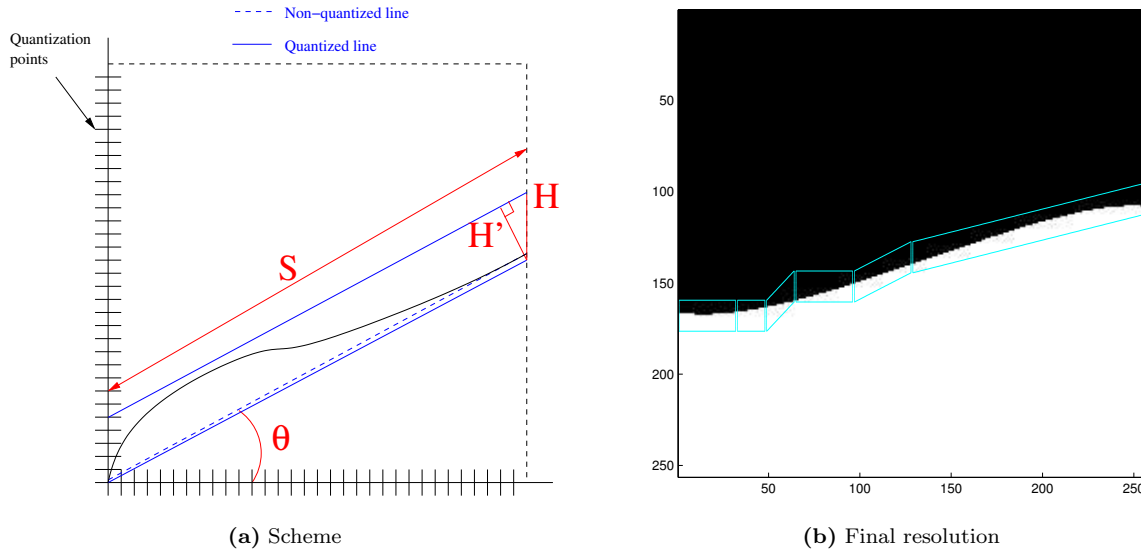


Figure 3.2: Anisotropic quadtree with rotation. (a) shows how the decomposition at one resolution level behaves, and (b) shows the final decomposition results when using 4 decomposition levels.

3.7 Anisotropic quadtree with rotation

Looking at Figure 3.1(b), one clearly sees that introducing anisotropy slightly reduces the number of “boxes” in a decomposition, but not impressively. Intuitively, the next step is to use not only rectangular, but rectangular and rotated boxes. The implementation of such a scheme looks difficult, but it can be achieved by a very simple procedure:

Taking the curve in the unit interval $[0, 1]^2$, one has to join the two extreme points with a line that represents the curve average slope. The end points of the line have to fall on a quantization point (see Figure 3.2(a)). This line will then be moved up and down, to form the upper limit and the lower limit of the partition in such a way that the edge never goes out of the partition, but that the partition limits are as close to the edge as the quantization resolution allows, forming a parallelogram that contains the edge.

The second resolution is obtained by splitting the x axis dyadically, and performing the same procedure inside every one of the new partitions. Following this scheme strictly, the next partition may not be completely nested to the previous one, but there will be either the upper corner or the lower corner of both the left and the right sides of the next partition that will be inside the previous one. One can slightly change the scheme by forcing to have nested partitions. This is done simply by slightly adjusting the slope of the next partition so that it does not fall outside the previous one. Forcing nested partitions does not change the number of terms or distortion evolution.

Similarly to the anisotropic quadtree algorithm, the y axis partition will basically depend on the edge, while the x axis partition depends on the iteration number. Fig. 3.2(b) represents this edge analysis.

3.7.1 Number of terms

Similarly to the anisotropic quadtree, the anisotropic and rotated quadtree is going to stop when the height H of the parallelogram (see Figure 3.2(a)) achieves the smallest resolution, which is 2^{-J}

if J refinement levels are allowed.

To be able to count the number of terms in the decomposition, the edge function inside every box will be approximated by its second order Taylor expansion at the central point. The first partition is the whole unit interval.

Defining x_- as the lowest point in the x axis which is inside the interval to analyze and x_+ as the highest one, the coordinates of the two extreme points of the curve quantized on a dyadic grid will be $(x_-, Q[h(x_-)])$ and $(x_+, Q[h(x_+)])$. The line which joins these two points is:

$$r(x) = Q[h(x_-)] + \frac{Q[h(x_+)] - Q[h(x_-)]}{x_+ - x_-}(x - x_-), \quad (3.40)$$

where $Q[\cdot]$ stands for uniform quantization. The parallelogram cannot cross the edge, therefore its superior and inferior distances to the line are, respectively:

$$d_+ = \max \left\{ 0, \sup_{x \in X_j} (h(x) - r(x)) \right\} \geq 0, \quad (3.41)$$

and

$$d_- = \max \left\{ 0, \sup_{x \in X_j} (r(x) - h(x)) \right\} \geq 0, \quad (3.42)$$

where X_j is the set of points in the x axis that belong to the partition under analysis. Then the height of the parallelogram confining the edge in partition j will be:

$$H_j = Q[d_+ + d_-]. \quad (3.43)$$

And the number of terms can be found by counting all the rectangles that achieve a height $H = 2^{-J}$ all over the quadtree depths. Knowing the expression of the height of the parallelogram at iteration j , one can compute the number of refinements necessary to achieve $H_j = 2^{-J}$ and thus find the number of edge partitions. Three cases have to be considered: d_+ and d_- are both greater than zero, one of them is equal to zero, and finally $d_+ = d_- = 0$.

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

To compute the height of one partition, one has to compute the point where the distance between the edge and the line $r(x)$ is maximum above (maximum positive distance) and below (maximum negative distance). Taking x_{d_+} as the point in the x axis where $r(x) - h(x)$ is maximum, and approximating the curve by its second order Taylor expansion in the central point of the interval being analyzed, we get:

$$\begin{aligned} d_+ &= h(x_{d_+}) - r(x_{d_+}) = \\ &= h\left(\frac{x_+ + x_-}{2}\right) + h'\left(\frac{x_+ + x_-}{2}\right)\left(x_{d_+} - \frac{x_+ + x_-}{2}\right) + \frac{1}{2}h''\left(\frac{x_+ + x_-}{2}\right)\left(x_{d_+} - \frac{x_+ + x_-}{2}\right)^2 \\ &\quad - h(x_-) - \frac{h(x_+) - h(x_-)}{x_+ - x_-}(x_{d_+} - x_-) + O\left(\left(x_{d_+} - \frac{x_+ + x_-}{2}\right)^3\right), \end{aligned} \quad (3.44)$$

where $r(x)$ has been substituted by its expression taking into account the quantization errors of $\frac{2^{-J}}{2}$, as J refinement steps are allowed:

$$r(x) = h(x_-) \pm \frac{2^{-J}}{2} + \frac{h(x_+) - h(x_-) \pm 2^{-J}}{x_+ - x_-}(x - x_-). \quad (3.45)$$

Taking Eq. (3.44) and substituting $h(x_-)$ and $h(x_+)$ by its Taylor expansion in the interval's middle point, gives:

$$\begin{aligned} d_+ &= \left[h' \left(\frac{x_+ + x_-}{2} \right) - \frac{h(x_+) - h(x_-)}{x_+ - x_-} \right] (x_{d_+} - x_-) \pm 2^{-J} \left(\frac{x_{d_+} - x_-}{x_+ - x_-} \right) \\ &+ \frac{1}{2} h'' \left(\frac{x_+ + x_-}{2} \right) \left[\left(x_{d_+} - \frac{x_+ + x_-}{2} \right)^2 - \left(\frac{x_+ - x_-}{2} \right)^2 \right] \\ &\pm \frac{2^{-J}}{2} + O \left(\left(x_{d_+} - \frac{x_+ + x_-}{2} \right)^3 \right) + O \left(\left(\frac{x_- - x_+}{2} \right)^3 \right), \end{aligned} \quad (3.46)$$

It is possible to show that [156, 157]:

$$\left| h' \left(\frac{x_+ + x_-}{2} \right) - \frac{h(x_+) - h(x_-)}{x_+ - x_-} \right| \leq 2^{-2j} \quad (3.47)$$

and so

$$\left[h' \left(\frac{x_+ + x_-}{2} \right) - \frac{h(x_+) - h(x_-)}{x_+ - x_-} \right] (x_{d_+} - x_-) \sim O(2^{-3j}). \quad (3.48)$$

It is also easy to see that:

$$\left| \left(x_{d_+} - \frac{x_+ + x_-}{2} \right)^2 - \left(\frac{x_+ - x_-}{2} \right)^2 \right| \leq 2^{-2(j+1)}, \quad (3.49)$$

which brings the following bound for the 3rd term in Eq. (3.46):

$$\frac{1}{2} h'' \left(\frac{x_+ + x_-}{2} \right) \left[\left(x_{d_+} - \frac{x_+ + x_-}{2} \right)^2 - \left(\frac{x_+ - x_-}{2} \right)^2 \right] \leq \frac{1}{2} \left| h'' \left(\frac{x_+ + x_-}{2} \right) \right| 2^{-2(j+1)}. \quad (3.50)$$

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

$$K_j^k = \left| h'' \left(\left(k + \frac{1}{2} \right) 2^{-j} \right) \right|, \quad (3.51)$$

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

$$\beta = \max_{0 \leq k \leq 2^j - 1} \left| h'' \left(\left(k + \frac{1}{2} \right) 2^{-j} \right) \right| < \infty. \quad (3.52)$$

From equation (3.46) it follows that at iteration j and in box k the asymptotic behavior of d_+ is given by:

$$d_+ \sim \frac{1}{2} K_j^k 2^{-J - \log_2 \beta} + \frac{3}{2} 2^{-J}. \quad (3.53)$$

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 exactly the same method and has an identical behavior.

The iterative algorithm is going to stop when the requested resolution is reached, *i.e.*, when $H_j = 2^{-J}$. Substituting in Eq. (3.43), the number of iterations needed to reach this parallelogram height can be obtained as a function of the resolution and of the maximum curvature of the edge:

$$J_{\text{stop}} = \max \left\{ 0, \frac{1}{2} (J + \log_2 \beta + 1) \right\}. \quad (3.54)$$

Thus, for this case, the number of non-zero terms that play a role in the decomposition is, at most, of $2^{J_{\text{stop}}}$:

$$N_j \leq \max \left\{ 1, 2^{\frac{1}{2}(J + \lceil \log_2 \beta \rceil + 1)} \right\} \sim 2^{\frac{J}{2}}. \quad (3.55)$$

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

This case turns out to have approximately the same number of terms than the previous one. Only some constants differ, but the general behavior is the same, so again:

$$N_j \leq \max \left\{ 1, 2^{\frac{1}{2}(J + \lceil \log_2 \beta \rceil + 1)} \right\} \sim 2^{\frac{J}{2}}. \quad (3.56)$$

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. If $d_+ = d_- = 0$, the parallelogram height will be:

$$H = 2^{-J}, \quad (3.57)$$

thus the number of non-zero coefficients turns to be one:

$$N_J = 1. \quad (3.58)$$

3.7.2 Distortion

The computation of the distortion in the anisotropic quadtree with rotation must, again, consider different cases, depending on the values of d_+ and d_- , described in Eqs. (3.41) and (3.42).

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

As in the other schemes, the final distortion of the anisotropic quadtree with rotation is the distortion at the finest resolution ($j = J_{\text{stop}}$). Taking the values obtained in the previous section, one can easily see that it takes the following form:

$$\begin{aligned} D &= \sum_{k=0}^{2^{J_{\text{stop}}-1}} (d_+ + d_-) 2^{-J_{\text{stop}}} = \sum_{k=0}^{2^{J_{\text{stop}}-1}} \left(K_{J_{\text{stop}}}^k + 3 \right) 2^{-3J_{\text{stop}}} \\ &= 2^{-2J_{\text{stop}}} \left(\sum_{k=0}^{2^{J_{\text{stop}}-1}} K_{J_{\text{stop}}}^k 2^{-J_{\text{stop}}} + \sum_{k=0}^{2^{J_{\text{stop}}-1}} 3 \cdot 2^{-J_{\text{stop}}} \right), \end{aligned} \quad (3.59)$$

where the one term distortion $(d_+ + d_-) 2^{-J_{\text{stop}}}$ (corresponding the area of one parallelogram) has been summed all over the terms that play a role in the decomposition to obtain the total one. In the right-hand side of equation (3.59), the second sum gives a constant, while the first, as $K_{J_{\text{stop}}}^k$ is related with the maximum value of the second derivative of the curve. When $J_{\text{stop}} \rightarrow \infty$, this term converges to the Riemann integral of the second derivative of the curve:

$$\lim_{j \rightarrow \infty} \sum_{k=0}^{2^{J_{\text{stop}}-1}} K_{J_{\text{stop}}}^k \cdot 2^{-J_{\text{stop}}} = \int_0^1 |h''(x)| dx. \quad (3.60)$$

This can be seen as an approximation of the total variation TV of the edge (Riemann integral of the curvature), with the only difference of having a sum of K_j^k instead of the integral of the curvature. Calling it \widetilde{TV} the final expression for the distortion becomes:

$$D \sim \left(\widetilde{TV} + 3 \right) \cdot 2^{-2J_{\text{stop}}} \sim \left(\widetilde{TV} + 3 \right) \cdot 2^{-J - \log_2 \beta}. \quad (3.61)$$

Remember that β corresponds to a measure of the maximum curvature, thus, the distortion of this scheme will be highly dependent on the geometry of the curve. Similarly to what happens with the number of non-zero terms, the more complex the curve, the bigger the distortion.

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

This case turns to have the same distortion as the previous one. The evolution of the distortion with the refinement does not change. Some constants differ, but the general behavior is the same.

Case $d_- = d_+ = 0$

The computation of the distortion for this case is very easy, as only one parallelogram with height 2^{-J} and width 1 is present here. The final distortion is thus:

$$D \leq 2^{-J} \quad (3.62)$$

3.7.3 Rate-Distortion

As it has been done for the study of the distortion and the number of terms in the decomposition, the study of the anisotropic quadtree with rotation requires three different cases:

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

Each rotated box is coded by means of H_j and a left and a right vertex, so at the first iteration J bits are needed to code H_0 and $2J$ bits for the position of the two vertices, which makes $3J$ bits. At iteration j , as at least two of the vertices of the following parallelogram will be inside the previous one, the number of bits needed to code one vertex of the box k depends on H_{j-1} as follows:

$$N_{\text{bits}_V}^k = \log_2 \frac{H_{j-1}^{k'}}{2^{-j}} = J - 2(j-2) + \lceil \log_2 (K_{j-1}^k) \rceil, \quad (3.63)$$

where k' is the index for the boxes of the previous iteration. As the number of bits needed to code all the boxes at iteration j is given by the bits for the two vertices and the height multiplied by the number of parallelograms, the total rate will be:

$$R = \sum_{j=0}^{J_{\text{stop}}} (2N_{\text{bits}_V} + N_{\text{bits}_H}) \times 2^j, \quad (3.64)$$

where N_{bits_H} is the number of bits needed to code the height of each box, which is the same as N_{bits_V} . So the final rate turns out to be:

$$\begin{aligned} R &= 3J+2 + \sum_{j=1}^{J_{\text{stop}}} \sum_{k=0}^{2^j-1} ((J-2(j-1) + \lceil \log_2 K_{j-1}^k \rceil) \cdot 3+2) \leq \\ &\leq 3J+2 + \sum_{j=1}^{J_{\text{stop}}} \sum_{k=0}^{2^j-1} ((J-2(j-1) + \lceil \log_2 \beta \rceil) \cdot 3+2) = \\ &= 3J+2 + \sum_{j=1}^{J_{\text{stop}}} ((J-2(j-1) + \lceil \log_2 \beta \rceil) \cdot 3+2) \cdot 2^j. \end{aligned} \quad (3.65)$$

This is an arithmetico-geometrical progression, whose sum is [83]:

$$R \leq 28 \cdot 2^{\frac{1}{2}(J + \lceil \log_2 \beta \rceil)} - 3J - 26 - 6 \lceil \log_2 \beta \rceil. \quad (3.66)$$

Finally, for J large enough, the dominant term is the first one, thus:

$$R \sim 2^{\frac{1}{2}(J + \lceil \log_2 \beta \rceil)}. \quad (3.67)$$

Combining this equation with (3.61) we obtain the asymptotic Rate-Distortion behavior:

$$D(R) \sim (\widetilde{\text{TV}} + 3) \cdot 2^{-2 \log R} \quad (3.68)$$

and so:

$$D(R) \sim (\widetilde{\text{TV}} + 3) \cdot R^{-2}. \quad (3.69)$$

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

This case turns out to have the same distortion rate as the previous one, because the evolution of the distortion with the refinement does not change. Only some constants differ, but the general behavior is the same.

Case $d_- = d_+ = 0$

This case is very favorable to our coding scheme, because with just one iteration the minimum distortion requirement is reached. As there is only one box, the rate is just the cost in bits to code the two vertices and the box's height:

$$R = 3J. \quad (3.70)$$

This makes a Rate-Distortion behavior of:

$$D(R) = 2^{-\frac{R}{3}}, \quad (3.71)$$

which is the optimal parametric rate, and is coherent with the results obtained in [42].

3.8 Anisotropic quadtree with rotation and refinement

The anisotropic quadtree with rotation and refinement uses the same division process as the anisotropic quadtree with rotation, but with an extra step: Once the smallest resolution has been achieved, it codes where the edge crosses the parallelogram left and right borders, in order to give a continuous edge line.

3.8.1 Number of terms

As in the case of the isotropic quadtree with refinement, the number of non-zero terms of the anisotropic quadtree with refinement is the same than the number of the non-zero terms of the anisotropic quadtree without refinement, because the number of partitions does not change, just the information every coefficient gives is different.

3.8.2 Distortion

As in the previous cases, the fact of adding the refinement in the scheme reduces the distortion of the final decomposition. Being L the number of levels of refinement allowed in a quadtree partition, the final distortion in this kind of schemes is as follows.

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

Following the same procedure, the distortion found for the case $d_+ > 0$ and $d_- > 0$ is:

$$D = \widetilde{\text{TV}} \cdot 2^{-2J} + 3 \cdot 2^{-J-L}, \quad (3.72)$$

where J is, as usual, the depth of the quadtree and L is the depth of the refinement in the finest quadtree decomposition level.

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

As usual, this case has just some minor changes with respect to the previous one, so it will not be recomputed here. The final distortion becomes:

$$D = \widetilde{\text{TV}} \cdot 2^{-2J} + 3 \cdot 2^{-J-L}, \quad (3.73)$$

where J is the depth of the quadtree and L is the depth of the refinement in the last quadtree decomposition level.

Case $d_- = d_+ = 0$

The distortion for this case, obtained with the same procedure as in the other cases, is:

$$D \leq 2^{-J-L} \quad (3.74)$$

3.8.3 Rate-Distortion

This section studies what the effect of the refinement in the Rate-Distortion behavior of the anisotropic quadtree with rotation. As usual, three cases are distinguished:

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

The anisotropic quadtree with refinement has a rate composed of two terms: the first term is the number of bits needed to code the position and the size of the parallelograms at every iteration (which is exactly the same as in Eq. (3.64)) plus the refinement bits.

The number of bits spent to refine depends on the finest resolution rectangle height. As the width of the last resolution parallelograms is known ($2^{-J_{\text{stop}}}$), it is easy to compute the number of refinements inside when refining every 2^{-J} as:

$$\#_{ref} = \frac{2^{-J_{\text{stop}}}}{2^{-J}} + 1 = 2^{\frac{1}{2}(J - \log_2 \beta)} + 1 \quad (3.75)$$

So, the final rate will be:

$$R = \sum_{j=0}^{J_{\text{stop}}} (2N_{bits_edge} + N_{bits_H}) \times 2^j + \left(L \cdot (2^{\frac{1}{2}(J - \log_2 \beta)} + 1) \right) \times 2^{J_{\text{stop}}}, \quad (3.76)$$

which, when evaluating the sum and substituting J_{stop} by its value (Eq. 3.54) becomes:

$$R = 2^{\frac{1}{2}(J + \log_2 \beta)} + L \cdot 2^J. \quad (3.77)$$

The final Distortion-Rate expression can be deduced from equations (3.72) and (3.77), and is again obtained considering $L = J$:

$$D(R) \sim \widetilde{\text{TV}} \cdot \frac{\log R}{R^2}. \quad (3.78)$$

As in the previous case, the distortion decreases as R^{-2} and directly depends on the curvature of the edge.

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

This case has the same distortion rate as the previous one, because the evolution of the distortion with the refinement does not change. Only some constant values differ, but the general behavior is the same.

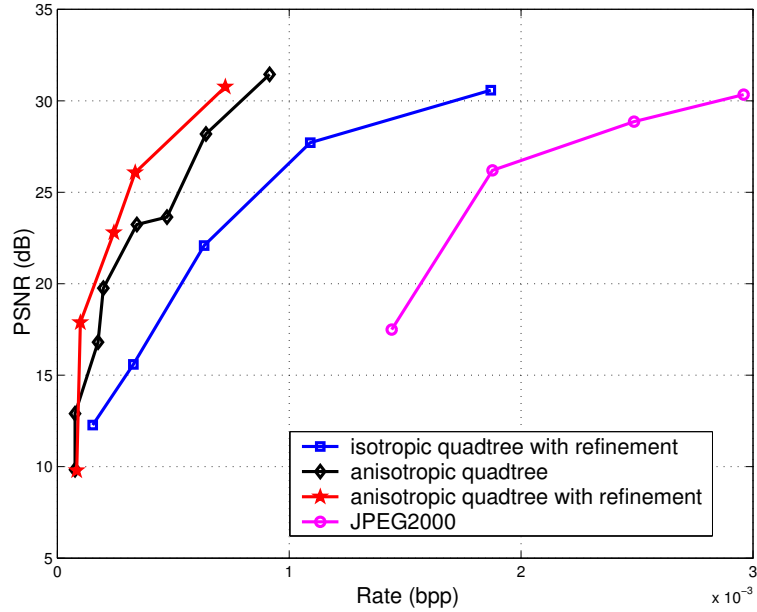


Figure 3.3: Comparison among JPEG2000, isotropic quadtree with refinement and anisotropic quadtree with rotation for an image of 1024×1024 pixels.

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. If $d_+ = d_- = 0$, the parallelogram width will be:

$$(d_+ + d_-) = 2^{-J-L}, \quad (3.79)$$

and the rate will be just the bits needed to code the two vertices and the parallelogram width:

$$R = 3J + L. \quad (3.80)$$

This, when $J=L$, makes a Rate-Distortion behavior of:

$$D(R) = 2^{-\frac{R}{2}}, \quad (3.81)$$

which is coherent with the results obtained in [42].

3.9 Practical results

Some comparisons among the anisotropic quadtree with rotation, the isotropic quadtree with refinement and wavelets (JPEG2000 [2]) are presented in Figure 3.3. 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. The JPEG2000 is far away to the right of the graph (it gives worse approximation), but this is due to the fact that it is adapted to natural images and not to black and white ones.

Fig. 3.4 represents the rate distortion decay of four different curves with increasing Total Variation. It shows that the practical results are coherent with the theoretical behavior: the lower the TV, the better the R-D. From left to right, the graph represents the rate-distortion of 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.

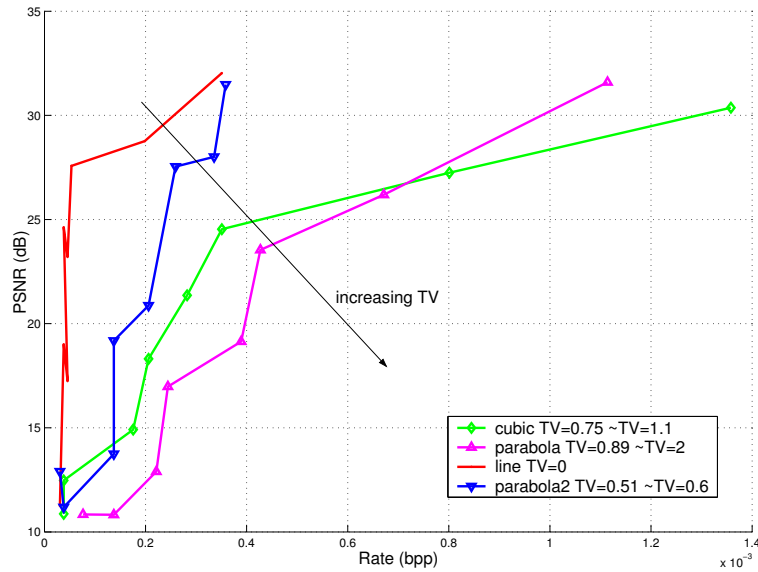


Figure 3.4: Comparison of the bit-rate of different curves with different Total Variation.

3.10 Conclusions

This chapter has performed a toy model for the study of the benefits of introducing rotation and anisotropy in the basis functions, in order to see if it is worth the effort of working with redundant dictionaries. Through a toy Horizon model image, a study about the number of basis functions needed to achieve a certain resolution level has been performed. This study has shown that introducing anisotropy and rotation in the basis functions allows to approximate a signal with fewer terms. A Rate-Distortion study has been performed as well, for this toy model, to see if there is a benefit in terms of Rate-Distortion in performing a coding scheme with redundant dictionaries that include anisotropy and rotation. The main conclusions of this study are summarized in Table 3.10. It has been seen that anisotropy and rotation not only allow to approximate a signal with fewer terms, but that the Rate-Distortion curve behaves better as well. Thus, it is worth studying the implementation of an image coder that uses redundant dictionaries with anisotropic and rotated functions. It is also interesting to notice that the addition of the refinement step in the anisotropic quadtree with rotation does not improve the Rate-Distortion behavior. This comes from the fact that, contrarily to the other schemes presented in this chapter, the anisotropic quadtree with rotation already follows the slope of the contour in its partitions. Refining this, thus, slightly improves the distortion, but not a lot. While in the isotropic quadtree and in the anisotropic quadtree, adding refinement implies adding a sort of rotation, in the anisotropic quadtree with rotation, this last parameter is not needed, due to the fact that this rotation is already taken into account in the construction of the partitions. The main improvement that refinement brings is that it allows to keep the continuity of the edge function.

Scheme	Number of terms	Distortion	Rate	R-D ($L = J$)
Isotropic Quadtree	$\sim 2^J$	2^{-J}	$J \cdot 2^J$	$\frac{\log R}{R}$
Optimized Quadtree	$\sim 2^J$	2^{-J-L}	$L \cdot 2^J$	$\frac{\log R}{R^2}$
Anisotropic Quadtree	$\leq 2^J$	2^{-J}	2^J	R^{-1}
Anisotropic Quadtree with Refinement	$\leq 2^J$	2^{-J-L}	$L \cdot 2^{J+\lceil \log_2 h' \rceil}$	$\frac{2^{2\lceil \log_2 h' \rceil} \log R}{R^2}$
Anisotropic Quadtree with Rotation	$\leq 2^{\frac{J}{2}}$	2^{-J}	$2^{\frac{1}{2}(J+\lceil \log_2 \beta \rceil)}$	$\frac{(\widetilde{\text{TV}}+3)}{R^2}$
Anisotropic Quadtree with Rotation and Refinement	$\leq 2^{\frac{J}{2}}$	2^{-J-L}	$2^{\frac{1}{2}(J+\lceil \log_2 \beta \rceil)} + L \cdot 2^J$	$\widetilde{\text{TV}} \cdot \frac{\log R}{R^2}$

Table 3.1: Comparison of the performance of different quadtree schemes in number of terms, distortion, rate and rate-distortion, for J quadtree decomposition levels and, if it proceeds, L levels of refinement.

Chapter 4

Sparse signal representations

The previous chapters have shown that arbitrary anisotropy and rotation in the basis functions helps getting sparser approximations. But the set of basis functions containing anisotropy and rotations forms a redundant dictionary. Working with a redundant set of basis functions has the inconvenient of having a non-unique signal approximation. We would like to recover, from all the possible approximations, the sparsest one. Finding the sparsest signal approximation in a redundant set of basis functions is NP-hard, and thus some techniques have to be used that allow to solve this problem. But these techniques do not always guarantee to be able find the optimal solution. This chapter reviews some of these techniques, and discusses their performances.

This chapter is structured as follows: Section 4.1 introduces the problem of sparse image representations. Section 4.2 gives some useful definitions to better understand the notations. Section 4.3 briefly introduces some of the most common algorithms for dealing with redundant dictionaries. Section 4.4 gives some conclusions about the properties of dictionaries that ensure recoverability of sparse signal approximations and dictionaries that are capable of sparsely approximating an image. Finally, Section 4.5 gives some conclusions about these dictionaries and their usefulness for sparse signal representation.

4.1 Introduction

Since the growing interest for sparse signal representations, redundant dictionaries have been more and more studied. Indeed, several authors have been trying to find which cases ensure that a given algorithm finds the sparsest possible solution in a fixed redundant dictionary. This implies two main issues: the redundancy of the dictionary of functions and the algorithm used to compute the sparse approximation.

In order to better understand what sparsity means, a proper definition about sparse representation should be given. The sparsest possible representation of a signal \mathbf{s} in an overcomplete dictionary \mathcal{D} with synthesis matrix Φ is the sequence of coefficients c that solves the optimization problem:

$$\min_c \|\mathbf{c}\|_0 \quad \text{subject to} \quad \mathbf{s} = \Phi \mathbf{c}, \quad (4.1)$$

where $\|\cdot\|_0$ is the ℓ^0 pseudo-norm (that counts the non-zero terms). When dealing with sparse approximations, the above minimization problem transforms to:

$$\min_c \|\mathbf{e}\|_2 \quad \text{subject to} \quad \mathbf{s} = \Phi \mathbf{c} + \mathbf{e} \quad \text{and} \quad \|\mathbf{c}\|_0 \leq m, \quad (4.2)$$

or, if what is fixed is not the number of terms but the maximum error allowed:

$$\min_c \|\mathbf{c}\|_0 \quad \text{subject to} \quad \mathbf{s} = \Phi \mathbf{c} + \mathbf{e} \quad \text{and} \quad \|\mathbf{e}\|_2 \leq \epsilon, \quad (4.3)$$

where $\epsilon \geq 0$. One may want to use a combination of the last two problems, in order to have a compromise between the error decay and the increasing number of terms:

$$\min_c \{ \|\mathbf{s} - \Phi \mathbf{c}\|_2^2 + \tau^2 \|\mathbf{c}\|_0 \}. \quad (4.4)$$

The sparse optimization problem defined in Eq. (4.4) is a non-convex discrete optimization problem, and thus cannot determine a global extrema with standard optimization algorithms. There are several different techniques that try to solve this optimization problem. Some of them, such as Basis Pursuit (BP) [21], try to transform this non-convex algorithm into a convex problem, and solve it by linear programming. Other techniques, such as Matching Pursuit (MP) [119] or Orthonormal Matching Pursuit (OMP) [143], try to solve it term by term, instead of solving the global problem. The most common algorithms that are used for sparse image representation and/or approximation will be explained in the next section.

Sparse approximation has been studied for a long time. Temlyakov [185] locates the first example in an article published at the beginning of the twentieth century [165]. Even though in the 1950s there were some investigations about subset selection in regression, reviewed in [128], it is not until the end of the twentieth century that a systematic study of the m -term representation with respect to orthonormal bases and redundant systems was started [76, 87, 88, 89, 108, 117, 117, 132]. The study of the sparsest approximation is even more complex than the study of sparsest signal representation, and it is also quite recently that it has been studied in depth [34, 77, 86, 90, 110, 185, 188, 189, 190, 191].

There are several questions to answer when looking for a sparse signal representation. First of all, the set of basis functions used is able to perform this sparse representation has to be found. In order to be able to perform a sparse representation of a signal, the considered set of basis functions has to be able to represent this signal with a number of terms that is smaller than the dimension of the signal. This is a very hard condition, and it was demonstrated that the probability of finding a natural signal that can be sparsely represented [188] forms a set of Lebesgue measure zero in \mathbb{C}^d . Nevertheless, there are situations where a signal can be designed with this exact sparse representation (such as in some communication problems), and thus, these results can be directly applied. However, the problem to focus for natural images is not to find a dictionary that can exactly represent a signal in a sparse way, but to find a dictionary that can efficiently approximate a signal with a small number of terms. As seen in the previous chapter, such a dictionary forms a redundant set of basis functions.

4.2 Some definitions

Before starting the description of different algorithms to recover sparse signal representation/approximation, it is worth giving some definitions. Let

$$\mathcal{D} = \{g_\gamma : \gamma \in \Gamma\} \quad (4.5)$$

be a redundant dictionary of basis functions, with Γ the set of basis function indices that generate the dictionary. Φ is the synthesis matrix of the dictionary (the matrix that maps coefficient vectors to signals). A signal \mathbf{s} will then be represented as:

$$\mathbf{s} = \Phi \mathbf{c} + \mathbf{e}, \quad (4.6)$$

where \mathbf{c} is the vector of coefficients and \mathbf{e} is the representation error, which will be zero when dealing with exact representations. Let

$$\mu := \max_{\substack{k \neq l \\ \gamma_k, \gamma_l \in \Gamma}} |\langle \mathbf{g}_{\gamma_k}, \mathbf{g}_{\gamma_l} \rangle| \quad (4.7)$$

be the *coherence* of the dictionary. The coherence μ , for a dictionary of normalized atoms, is always $0 \leq \mu < 1$. The coherence is a measure of the maximum similarity of two basis functions in the dictionary: the smaller the coherence, the more different is one basis function of the dictionary from another. Notice that having μ near to one does not mean that all the basis functions of the dictionary are similar among them, it just means that there are two basis functions in the dictionary that are similar. To have a better measure of the coherence of a dictionary, it is more appropriate to use the *cumulative coherence* or *Babel function* [188, 189, 191]:

$$\mu_1(m, \mathcal{D}) := \max_{\text{card}(\Lambda) \leq m} \max_{\omega \notin \Lambda} \sum_{\lambda \in \Lambda} |\langle \mathbf{g}_{\lambda}, \mathbf{g}_{\omega} \rangle|. \quad (4.8)$$

Finally, M will be the number of terms used in the sparse representation.

4.3 Sparse signal representation with redundant dictionaries

Redundant dictionaries do not provide a unique signal representation, and finding the sparsest solution is not an easy task since performing a full search is an NP-hard problem [29, 134]. In this section, a brief explanation about the most common techniques used to solve this problem, with special focus on Matching Pursuit (which will be used throughout this thesis) and Basis Pursuit (which has brought interesting theoretical results during the last years) will be introduced. All the techniques presented here have one common point: they try to find a solution to the NP-hard problem of finding the sparsest signal approximation.

4.3.1 Basis Pursuit and Basis Pursuit Denoising

Basis Pursuit (BP) tries to solve the non-convex ℓ^0 minimization problem of Eq. (4.1) by solving instead the ℓ^1 norm convex problem [21]:

$$\min_c \|\mathbf{c}\|_1 \quad \text{subject to} \quad \mathbf{s} = \Phi \mathbf{c}. \quad (4.9)$$

But once again, this method is not applicable for real world signals, because the probability of having an exact sparse representation *forms a set of Lebesgue measure zero in \mathbb{C}^d* [188]. To overcome this problem inherent to BP, Basis Pursuit DeNosing (BPDN) is introduced [21]. Basis Pursuit denoising is a method that tries to solve the non-convex optimization problem of Eq. (4.4) by transforming it to a convex problem that uses the ℓ^1 norm instead of the ℓ^0 [194]:

$$\min_c \left\{ \frac{1}{2} \|\mathbf{s} - \Phi \mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_1 \right\}. \quad (4.10)$$

Generally speaking, Basis Pursuit is more a paradigm to solve the sparse signal representation problem than an algorithm. Several algorithms can be used to find the solution to the optimization

problem defined in Eq. (4.10). The advantage of Basis Pursuit is that one can control, with the λ parameter in Eq. (4.10), how much energy must the residual lose when adding a new coefficient. The disadvantage of BP is that one cannot directly control the number of terms that the minimization procedure will give. Modifying the λ parameter will certainly increase or decrease the ℓ^1 norm, but not necessarily the number of terms. It can happen that the number of terms stays constant, and that only the magnitude of the decomposition coefficient changes. On the contrary, it can also happen that with a little change in λ , the number of terms suddenly increases or decreases.

4.3.2 Greedy algorithms

Greedy algorithms were first introduced for solving statistical problems [70, 71]. They were used to deal with high dimensional data, when the observations of the random variable were sparse [70]. The theoretical background of greedy algorithms was first studied in [35], even though [99] had already performed a study of the convergence of greedy algorithms. Greedy algorithms iteratively construct an approximant by selecting the element of the dictionary that best matches the signal at each iteration. The pure greedy algorithm is known as *Matching Pursuit* [119], but there are variations of the Greedy algorithm, such as the Weak Greedy algorithm [182] with its variations [183, 186], or the Vector greedy algorithms [116].

Greedy algorithms offer constructive procedures for computing highly non-linear N -term approximations. Although the mathematical analysis of their approximation properties is difficult, due to their nonlinear nature, interesting results have been emerging (see for example [79]). The choice of a particular algorithm generally consists in trading off complexity and optimality, or more generally efficiency. The new image compression scheme presented in Chapter 6 proposes to use Matching Pursuit as a suboptimal algorithm to obtain a sparse signal expansion, yet an efficient way to produce a progressive low bit-rate image representation with a controlled complexity. Matching Pursuit, as already stressed before, iteratively chooses the best matching terms in a dictionary. Despite its possible numerical complexity, MP is very easy to implement. Moreover, since there is almost no constraint on the dictionary itself, MP clearly stands as a natural candidate to implement an efficient coding scheme based on anisotropic refinement, and such a construction is detailed in the next section.

Matching Pursuit

Let \mathbf{s} be a signal to represent in a given dictionary \mathcal{D} with synthesis matrix Φ . Assuming that all atoms in \mathcal{D} have unit norm, let the algorithm be initialized by setting $\mathbf{R}^0 \mathbf{s} = \mathbf{s}$. The signal is first decomposed as

$$\mathbf{R}^0 \mathbf{s} = \langle \mathbf{g}_{\gamma_0}, \mathbf{R}^0 \mathbf{s} \rangle \mathbf{g}_{\gamma_0} + \mathbf{R}^1 \mathbf{s}, \quad (4.11)$$

with \mathbf{g}_{γ_0} the function in the dictionary that minimizes the energy of the residual (equivalent to maximize the scalar product between the signal to represent and the basis functions \mathbf{g}_{γ_0}):

$$\mathbf{g}_{\gamma_0} \quad \text{s.t.} \quad |\langle \mathbf{R}^0 \mathbf{s}, \mathbf{g}_{\gamma_0} \rangle| = \sup_{\gamma \in \Gamma} |\langle \mathbf{R}^0 \mathbf{s}, \mathbf{g}_{\gamma} \rangle|, \quad (4.12)$$

Γ being the set of indices of the basis functions in the dictionary. Clearly \mathbf{g}_{γ_0} is orthogonal to $\mathbf{R}^1 \mathbf{s}$ and thus:

$$\|\mathbf{R}^0 \mathbf{s}\|_2^2 = |\langle \mathbf{g}_{\gamma_0}, \mathbf{R}^0 \mathbf{s} \rangle|^2 + \|\mathbf{R}^1 \mathbf{s}\|_2^2. \quad (4.13)$$

At the next step, the same procedure is applied to $\mathbf{R}^1 \mathbf{s}$, which yields to

$$\mathbf{R}^1 \mathbf{s} = \langle \mathbf{g}_{\gamma_1}, \mathbf{R}^1 \mathbf{s} \rangle \mathbf{g}_{\gamma_1} + \mathbf{R}^2 \mathbf{s}, \quad (4.14)$$

where \mathbf{g}_{γ_1} maximizes $|\langle \mathbf{g}_{\gamma_1}, \mathbf{R}^1 \mathbf{s} \rangle|$. Iterating M times this procedure, the following signal approximant is obtained:

$$\mathbf{s} = \sum_{m=0}^{M-1} \langle \mathbf{g}_{\gamma_m}, \mathbf{R}^m \mathbf{s} \rangle \mathbf{g}_{\gamma_m} + \mathbf{R}^M \mathbf{s}, \quad (4.15)$$

where the norm of the residual (approximation error) satisfies

$$\|\mathbf{R}^M \mathbf{s}\|^2 = \|\mathbf{s}\|^2 - \sum_{m=0}^{M-1} |\langle \mathbf{g}_{\gamma_m}, \mathbf{R}^m \mathbf{s} \rangle|^2, \quad (4.16)$$

with m the iteration number in which the basis function \mathbf{g}_{γ_m} has been chosen.

In [99] it was proved that Matching Pursuit converges and even converges *exponentially* in the strong topology in finite dimension (see [119] for a proof). Unfortunately, this is not true in the general case of infinite dimension, even though this property holds for particular dictionaries [198]. However, DeVore and Temlyakov [35] constructed a dictionary for which even a good signal, i.e., a sum of two dictionary elements, has a very bad rate of approximation. In this case a very sparse representation of the signal exists, but the algorithm dramatically fails to recover it! Notice though, as already stated before, that this does in no way rule out the existence of *particular* classes of efficient dictionaries. Section 4.4.3 gives an overview about under which conditions Matching Pursuit is capable of finding the sparsest approximation. Matching Pursuit does not generally achieve an exact representation with a finite number of terms. However, this can be solved by using Orthogonal Matching Pursuit, that recursively orthogonalizes the set of selected atoms.

Orthogonal Matching Pursuit

A clear drawback of the pure greedy algorithm is that the expansion of s on the linear span of the selected atoms is not the best possible one, since it is not an orthogonal projection. Orthogonal Matching Pursuit (OMP) [30, 143] solves this problem by recursively orthogonalizing the set of selected atoms using a Gram-Schmidt procedure [5]. The best M -term approximation on the set of selected atoms is thus computed and the algorithm can be shown to converge in a finite number of steps, but at the expense of a much bigger computational complexity.

The selection procedure of OMP is similar to MP:

$$\mathbf{R}^m \mathbf{s} = \langle \mathbf{g}_{\gamma_m}, \mathbf{R}^m \mathbf{s} \rangle \mathbf{g}_{\gamma_m} + \mathbf{R}^{m+1} \mathbf{s}. \quad (4.17)$$

Once the basis function has been chosen, the Gram-Schmidt algorithm orthogonalizes \mathbf{g}_{γ_m} with respect to $\{\mathbf{g}_{\gamma_p}\}_{0 \leq p < m}$ and defines:

$$\mathbf{u}_m = \mathbf{g}_{\gamma_m} - \sum_{p=0}^{m-1} \frac{\langle \mathbf{g}_{\gamma_m}, \mathbf{u}_p \rangle}{\|\mathbf{u}_p\|^2} \mathbf{u}_p. \quad (4.18)$$

The residue $\mathbf{R}^m \mathbf{s}$ is then projected on \mathbf{u}_m instead of \mathbf{g}_{γ_m} :

$$\mathbf{R}^m \mathbf{s} = \frac{\langle \mathbf{R}^m \mathbf{s}, \mathbf{u}_m \rangle}{\|\mathbf{u}_m\|^2} \mathbf{u}_m + \mathbf{R}^{m+1} \mathbf{s}. \quad (4.19)$$

The M term representation of s is then given by:

$$\mathbf{s} = \sum_{m=0}^{M-1} \frac{\langle \mathbf{R}^m \mathbf{s}, \mathbf{u}_m \rangle}{\|\mathbf{u}_m\|^2} \mathbf{u}_m + \mathbf{R}^M \mathbf{s}, \quad (4.20)$$

where $M < S$, with S the dimension of the signal space of s . If $M = S$, then

$$\mathbf{s} = \sum_{m=0}^{M-1} \frac{\langle \mathbf{R}^m \mathbf{s}, \mathbf{u}_m \rangle}{\|\mathbf{u}_m\|^2} \mathbf{u}_m. \quad (4.21)$$

Thus, convergence is obtained with a finite number of iterations (for signal with finite dimension).

It is possible to expand \mathbf{s} over the original dictionary vectors, and just change the magnitude of the projection coefficients through a change of basis. This can be done by inverting the triangular Gram-Schmidt relations in Eq. (4.18) to expand \mathbf{u}_m in $\{\mathbf{g}_{\gamma_p}\}_{0 \leq p < m}$:

$$\mathbf{u}_m = \sum_{p=0}^m b[p, m] \mathbf{g}_{\gamma_p}. \quad (4.22)$$

The M term approximation of the signal s can then be written as:

$$\mathbf{s}_M = \sum_{p=0}^{M-1} a[\gamma_p] \mathbf{g}_{\gamma_p}, \quad (4.23)$$

with

$$a[\gamma_p] = \sum_{m=p}^{M-1} b[p, m] \frac{\langle \mathbf{R}^m \mathbf{s}, \mathbf{g}_{\gamma_m} \rangle}{\|\mathbf{u}_m\|^2}. \quad (4.24)$$

This is often done, when using OMP decompositions, in order to avoid performing the Gram-Schmidt orthogonalization of the basis functions both in the decomposition and in the reconstruction process. The interested reader can find optimized versions of Orthogonal Matching Pursuit in [78, 152].

4.3.3 Weak Greedy algorithms

Even though greedy algorithms transform, for finite dictionaries, an NP-hard problem into a problem that has Polynomial complexity, it is still computationally hard to perform all the necessary scalar products when the dictionary of basis functions is large. This has motivated the study of Weak Greedy algorithms [35, 182]. Weak Greedy algorithms do not need to find the dictionary function that has the biggest projection. The atom selection rule is modified, allowing to choose a slightly suboptimal candidate:

$$\mathbf{g}_{\gamma_m} \quad s.t. \quad |\langle \mathbf{R}^m \mathbf{s}, \mathbf{g}_{\gamma_m} \rangle| \geq t_m \sup_{\mathbf{g} \in \mathcal{D}} |\langle \mathbf{R}^m \mathbf{s}, \mathbf{g}_{\gamma_{opt}} \rangle|, \quad t_m \leq 1. \quad (4.25)$$

Apart from the selection criteria, Weak Greedy algorithms work exactly as Matching Pursuit: they select the basis functions iteratively, from the approximation residual.

It was demonstrated in [184] that Weak Greedy algorithms are also able, when certain conditions hold, to converge to the optimal solution [184], even though the conditions under which this kind of algorithms converge to the optimal solution are more strict, and, thus, more difficult to be held. Weak Greedy algorithms are useful for problems that are too complex to solve even with the Greedy algorithm.

4.3.4 Other existing algorithms

There are other algorithms that try to perform sparse signal representations, even though some of them perform this kind of representations “unconsciously”. Most of them are not designed for a generic application, but focused on a concrete one. Some examples of these algorithms are:

FOCUSS (FOcal Underdetermined System Solver) [80] and its improved versions [130]. FOCUSS is a nonparametric algorithm for finding localized energy solutions from limited data. The FOCUSS algorithm consists of two parts: The first part finds a low resolution estimate of the sparse signal. The second part prunes this solution, using a generalized Affine Scaling Transformation (AST) to a sparse signal representation. The AST scales the entries of the current solution by those of the solutions of previous iterations. It finds the solution at each iteration by minimizing the ℓ^2 norm of the transformed variable. In some cases it is used to find a smaller dictionary that sparsely represents a signal [108].

Bayesian approach [187, 209], which tries to learn the sparse approximation of a signal in a Bayesian way from a probabilistic perspective.

Nonlinear programming methods [9], such as interior point method [170] or block coordinate relaxation [162]. These methods converge to the solution of BP.

Method of Frames [27], where sparsity is not defined as having a low number of terms, but as being localized in time and frequency.

Best Orthogonal Basis [24], where, from a predefined library of modulated waveforms, which include orthogonal wavelet-packets and localized trigonometric functions, builds an orthonormal basis relative to which the given signal or collection of signals has the lowest information cost. The cost function can be the ℓ^0 norm if one wants to achieve sparsity, or an entropy measure, if the resulting library is intended to be used for compression.

Approximation using coherence [79] is an interesting approach to achieve sparse signal approximation, but it can only be used for low coherence dictionaries.

Apart from the mentioned algorithms, a wide variety of algorithms that try to recover a sparse solution exist. Some of them, such as [134] try to merge two algorithms, in order to improve their behavior.

4.4 Recoverability of sparse signal approximations

All the algorithms presented in the previous section do not guarantee to find the sparsest solution defined in Eq. (4.2). As it is impossible to know in advance whether a given algorithm will give the sparsest possible approximation, several authors have studied their properties of approximation. In this section, the conditions under which three of the presented algorithms (BP, BPD and MP) achieve sparse signal approximation or representation will be presented. It will be seen that one of the main conditions to make sure that an algorithm finds the sparsest solution will be to have a dictionary with low coherence. Nevertheless, it has to be pointed out that, if the conditions of recoverability of the solution are not held, it does not necessarily mean that the algorithm is incapable of finding the sparsest solution. It just means that one cannot be sure that this is the solution that has been recovered.

4.4.1 Basis Pursuit

Basis Pursuit (BP) is an algorithm intended for exact sparse signal representation [47]. It considers that it is possible to represent a signal \mathbf{s} as:

$$\mathbf{s} = \Phi \mathbf{c}, \tag{4.26}$$

and it tries to find the sequence x of coefficients that has the minimum ℓ^1 norm (see Eq. (4.9)). BP has proved to be able to recover a signal formed by the union of two orthogonal bases (spikes and sinusoids) if this signal has an exact representation sparse enough. Sparse enough means:

$$M_t + M_\omega < \frac{\sqrt{N}}{2}, \quad (4.27)$$

where M_t is the number of spikes in the signal, M_ω is the number of non-zero sinusoid coefficients, and N is the dimension of the signal space. This is a very concrete case, and it does not ensure that BP is capable of finding the sparsest representation when using a more general dictionary. In [49], Donoho and Huo extend the result to any union of two bases. They proved that if the solution is very sparse in a dictionary formed by the union of two orthogonal bases, thus, if

$$M < \frac{1}{\mu}, \quad (4.28)$$

then, the solution is unique. BP can recover this unique solution for any union of two bases if:

$$M < \frac{1}{2} (1 + \mu^{-1}), \quad (4.29)$$

so, if the number of terms involved in this representation is inversely proportional to the coherence between the two bases. This condition was demonstrated to be sufficient and necessary [60], and improved [55, 56] to:

$$M < \frac{\sqrt{2} - 0.5}{\mu}. \quad (4.30)$$

In spite of the importance of these results, which show that BP can, in some cases, recover an exact sparse representation of a signal, the above results are only valid for dictionaries formed by the union of two orthogonal bases. [47] extends the results of uniqueness of sparse representations and recoverability with the ℓ^1 minimization, to general dictionaries. [88] is also an extension, but with special care to the case where the dictionary is formed by the union of L orthogonal bases. [76] proved that BP will always recover the good solution if the limit given in Eq. (4.29) is held. The generalization of sparse representations to any sparsity measure f (see Appendix A) is very interesting as well. In [87, 89], they generalized the uniqueness and coincidence of a maximally sparse solution for the ℓ^0 and ℓ^1 norm to a set of norms that are in between (such as ℓ^p , with $0 < p < 1$).

Basis Pursuit is able to recover an exact sparse decomposition. For real world signals, exact sparse representations are not possible, we have to deal with sparse approximations. BP cannot deal with sparse approximations, but Basis Pursuit Denoising (BPD) can.

4.4.2 Basis Pursuit Denosing

To study sparse approximations, one usually considers a signal that has an M -exact sparse representation, but with some additive noise (see Eq. (4.6)). Under these circumstances, Fuchs [77] demonstrated the following theorem:

Theorem 4.1 *Let $\mathbf{s} = \Phi \mathbf{c} + \mathbf{e}$ with $\|\mathbf{e}\|_2 \leq \epsilon$ and $M = \|\mathbf{x}\|_0 \leq \frac{\beta}{\mu}$ with $\beta \leq \frac{1}{2}$. Then, if one solves Eq. (4.10) with*

$$\lambda > \left(1 + \frac{\beta}{1 + \mu - 2\beta} \right) \epsilon \quad (4.31)$$

the support of the optimum $c_1(h)$ that solves Eq. (4.9) is either identical or contained in the support of x . And if one solves Eq. (4.9) with

$$\lambda = \left(1 + \frac{\beta}{1 + \mu - 2\beta}\right) \epsilon \quad (4.32)$$

and if

$$\min_i |c_i| > \left(1 + \frac{1}{1 + \mu - 2\beta}\right) \epsilon, \quad (4.33)$$

with $\min_i |c_i|$ the smallest absolute value nonzero component of c , then:

$$\text{support } c = \text{support } c_1, \quad (4.34)$$

$$\text{sign } c = \text{sign } c_1. \quad (4.35)$$

This theorem basically means that if the difference between the energy of the residual and the minimum coefficient in the decomposition is big enough, the good set of basis functions has been found. Similar results were obtained by Tropp in [189].

4.4.3 Matching Pursuit and Orthogonal Matching Pursuit

Greedy algorithms have been widely used for a large number of applications, such as classification [147], stochastic signal processing [53], coding [135], etc. Matching Pursuit and Orthogonal Matching Pursuit were used even before there was any theoretical proof that could provide the sparsest solution, just because they guarantee that the energy of the error always decreases. In the last years, there has been a great interest in proving the convergence and the sparsest approximation properties of greedy algorithms, and several authors have presented studies on this issue [36, 79, 91, 104, 105, 106, 188] and some of its variations [111, 112, 116]. This section will try to summarize the capability of recovering the sparsest approximation with MP and OMP, and to give an overview of the approximation properties of Greedy algorithms.

The first convergence results for MP and OMP were only valid for dictionaries with a certain structure, as in [198, 199], where Villemoes demonstrated that MP converged exponentially when using a Walsh atom dictionary. Temlyakov *et al.* also studied the convergence of Greedy algorithms in Banach spaces with certain smoothness assumptions and in the trigonometric system [36, 105, 106, 183] and even for general spaces [104]. These works ensure that the greedy algorithms solution converges or that the difference of the optimal solution and the real one has bounded energy, but they do not guarantee that Greedy algorithms are able to recover the sparsest signal approximation. The fact that some Greedy algorithms are able to recover the sparsest signal approximation is pointed out in [188, 191], with the following theorem concerning Orthogonal Matching Pursuit:

Theorem 4.2 *For any signal \mathbf{s} , Orthogonal Matching Pursuit can calculate an approximant \mathbf{a}_K that consists of $0 \leq K \leq M$ atoms from the optimal M -term approximant \mathbf{a}_{OPT} and that satisfies the error bound:*

$$\|\mathbf{s} - \mathbf{a}_K\|_2^2 \leq \left[1 + \frac{M(1 - \mu_1(M))}{(1 - 2\mu_1(m))^2}\right] \|\mathbf{s} - \mathbf{a}_{OPT}\|_2^2. \quad (4.36)$$

This result is generalized for MP in [91], where it is proved that if the signal atoms are incoherent enough with the rest of the dictionary, MP always recovers the good expansion:

Theorem 4.3 *Let \mathcal{D} be a dictionary in a finite or infinite dimensional Hilbert space and I an index set such that the Stability condition (SC) is met:*

$$\eta(I) := \sup_{k \notin I} \|(\Phi_I)^{-1} \mathbf{g}_{\gamma_k}\|_1 < 1, \quad (4.37)$$

with $(\cdot)^{-1}$ the pseudo-inversion. Then, for any $\mathbf{s} = \sum_{k \in I} c_k \mathbf{g}_{\gamma_k} \in \text{span}(\mathbf{g}_{\gamma_k}, k \in I)$, MP:

1. picks up only “correct” atoms at each step ($\forall n$ s.t. $k_n \in I$).
2. if I is a finite set, then the residual r_m converges exponentially to zero.

The previous theorem, when dealing with incoherent enough dictionaries, is even more explicit, and says how many atoms can be picked and the degree of convergence of approximation of MP:

Theorem 4.4 *Let \mathcal{D} be a dictionary in a finite or infinite dimensional Hilbert space and μ its coherence. For any finite set I of size $\text{card}(I) = m < \frac{(1+\frac{1}{\mu})}{2}$ and any $\mathbf{s} = \sum_{k \in I} c_k \mathbf{g}_{\gamma_k} \in \text{span}(\mathbf{g}_{\gamma_k}, k \in I)$:*

1. MP picks up only “correct” atoms at each step ($\forall n, k_n \in I$)
2. converges exponentially

$$\|\mathbf{s}_n - \mathbf{s}\|^2 \leq \left(\left(1 + \frac{1}{\mu}\right) (1 + \mu) \right)^n \|\mathbf{s}\|^2 \quad (4.38)$$

Theorem 4.4 is important for two reasons: it limits the energy of the error and it tells how much terms a sparse signal can have in order to be recoverable through MP. Obviously, using MP one will not recover the exact value of the coefficients, which would be recovered when using OMP, but these can be easily recovered if one is sure to have recovered the good expansion of atoms.

4.5 Conclusions

There are several conclusions that can be drawn from this chapter. The first is the capability of recovering, in some cases, the sparsest signal approximation. All the theorems presented in this chapter concerning this capability reach the same conclusions: one can recover a sparse approximation or representation if the coherence of the dictionary in use is low enough. The contradictory point on this is that, in many cases, the dictionaries that do have a sparse signal approximation are highly coherent dictionaries. Thus, we can state the following conjecture:

Conjecture 4.1 (Uncertainty principle of sparse approximations) *There is a trade-off in sparse approximations of natural signals between the capacity of finding the sparsest approximation and the capacity of the dictionary to sparsely approximate the signal. The more capacity a dictionary has of sparsely approximating a signal, the more difficult it will be to recover this sparse approximation through a Polynomial complexity algorithm.*

For natural signals, it may be better to take an algorithm that guarantees that the error will converge to zero even if the sparsest approximation has not been found. However, it can be useful that this algorithm is able to recover this sparsest approximation if some conditions hold. Matching Pursuit and Orthogonal Matching Pursuit have these properties.

In addition, choosing a Greedy algorithm to try to find a sparse signal representation has the advantage of controlling directly the number of terms to be used, due to its iterative nature. Basis Pursuit and most of the other existing algorithms for sparse signal approximation do not allow to directly choose the number of terms, but have a trade-off between the approximation error and the number of terms.

Finally, from a numeric point of view, Matching Pursuit is an algorithm that is easy to implement and to use for different types of dictionaries of functions, and is less computationally hard than Orthogonal Matching Pursuit, even if it has slower approximation rate. As images are big data structures, the computational complexity of OMP or BP (even for their optimized implementations) limit somehow the usability of these algorithms for image representation. For all this, the algorithm that has been chosen in this thesis to try to find sparse image approximations is Matching Pursuit.

Chapter 5

An algorithm independent sparsity measure

5.1 Introduction

In the previous chapter, a review about the state of the art on sparse representations and approximations has been presented. The capability of a given algorithm to find the sparsest representation/approximation of a signal given a dictionary has been analyzed, trying to summarize the work of several authors working on this subject. It has been pointed out, as well, that most of the conditions that exist for verifying whether a representation is sparse or not are algorithm dependent or/and need data that is difficult to have, because it depends on the optimal sparse representation (such as the optimal number of terms, or the energy of the error, or the value of the smallest coefficient that belongs to the sparsest approximation).

This chapter*, based on [84, 85], presents a new sparsity test. This test allows to check whether a given sparse approximation is the sparsest possible or not. The main novelties of the test are:

- It is algorithm independent: the algorithm which has been used to obtain the signal approximation is no more a limitative factor for knowing the sparsity of the solution.
- It does not need any data from the optimal solution: all the data that are needed to perform the test depend on the solution that is being analyzed, the dictionary and the signal.
- It is, in most cases, tighter than previously existing tests.

This chapter is organized as follows: Section 5.2 presents the main differences between the work presented here and the work presented by other authors, and sets the goals and utilities of the test. Section 5.3 states the sparse approximation problem and introduces the main concepts and results. The meaning of the results, together with a discussion about the connections between sparse *models*,

*Work partly performed while the author of this thesis was in IRISA, Rennes, France, in collaboration with R. Gribonval.

sparse *optimization problems* and sparse approximation *algorithms* are discussed in this section. At the end of the section, some explicit examples to illustrate how one can use these results to build a numerical test of optimality of a sparse approximation are presented. The rest of the chapter is devoted to the proof of the results. In Section 5.4 some useful definitions and properties which are at the core of the proofs of the results are given. Section 5.5 contains the proofs for the case of the canonical basis in an ℓ^p space. Section 5.6 generalizes the previously introduced work, providing some abstract results for arbitrary dictionaries and general sub-additive distortion measures. At the end of this section, some examples show how these abstract results can be used to recover results from [48]. The main results with incoherent dictionaries in Hilbert spaces are proved in Section 5.7, where their sharpness is also discussed. Finally, Section 5.8 gives some conclusions.

5.2 Some remarks about the optimality test

As already mentioned in the previous chapter, recovering a sparse approximation of a signal is of great interest in many applications, such as coding [110], source separation [213] or denoising [178]. Chapter 4 presented several algorithms (Matching Pursuit [119, 143], Basis Pursuit [21], FOCUSS [80], etc.) that try to sparsely decompose a signal in a dictionary. It has been seen that it is generally difficult to prove whether a computed solution is the sparsest approximation we could obtain given a certain sparsity measure (which can be the number of terms or ℓ^0 “norm”, the ℓ^1 norm, or any other metric that may lie “in between” [87], related, for example, to the bit-rate needed to represent the coefficients). It would be interesting to have a tool for checking whether a certain solution is optimal or not. This tool should be algorithm independent, and should not depend on any data that can only be known if the optimal solution, or the number of terms that form the sparse representation are known. It should just say whether, for the computed number of terms, the solution found is the sparsest (the one that gives, for this number of terms, the smallest error) or not.

In this chapter, a general tool for checking that the solution computed by some algorithm is nearly optimal is provided. A solution will be considered to be optimal when no significantly different sparse linear expansion from the dictionary can provide both a smaller approximation error and a better sparsity. A positive test results basically implies that:

- For the number of terms of the computed solution, the sparsest solution is unique
- The decomposition found is the sparsest (or is very close to the sparsest) for this number of terms

The test quite naturally consists in checking that the residual (the difference between the signal and its sparse approximation) is “small enough” given the sparsity of the approximant and the magnitude of its smallest non-zero coefficient. When the test is satisfied, the computed solution is so close to the optimally sparse approximation –in the sense of the ℓ^0 norm– that there is an easy way to explicitly compute the latter.

The results presented here have several implications. They can be interpreted from two different levels of signification: from a numerical optimization point of view and from a modeling point of view.

From a numerical optimization point of view, when the test is satisfied, one knows for sure that the algorithm used to decompose the analyzed signal indeed solved a NP-hard problem. Since any reasonable person would use a polynomial time algorithm, this might seem contradictory at first sight (if $\text{NP} \neq \text{P}$), but it is not: the algorithm solved a *particular instance* of the NP-hard problem, but it will fail on at least one other instance of the problem.

From a modeling point of view, it is often reasonable to assume that most signals in a class of interest (audio signals, natural images, etc.) belong to a “good set” of instances where the NP-hard problem can be solved in polynomial time. This, which may seem to be an audacious data model, may be true if the data and the dictionary have certain features. Indeed, as a by-product of our results, if the analyzed signal can be modeled as the superposition of a “sufficiently sparse” component and a “sufficiently small” noise, then the “sufficiently sparse” component is close to the solutions of both an (*a priori* NP-hard) ℓ^0 -sparse approximation problem and a (convex) ℓ^1 -sparse approximation problem. Thus, the three of them (the noise, the ℓ^0 -sparse approximation and the ℓ^1 sparse approximation) can therefore be estimated in polynomial time by solving a convex optimization problem, as already proved by Donoho, Elad and Temlyakov [48], Tropp [189] and Fuchs [77], in their works on the topic of recoverability of sparse approximations and overcomplete representations. However, emphasis is put here on testing the near optimality of a computed sparse approximation, rather than predicting the recovery of an ideal sparse model with additive noise. Several other features distinguish this work from previous ones:

- previous results on recovery of sparse expansions in the noisy setting [48, 77, 91, 188, 189] make assumptions on the *ideal* sparse approximation which do not seem easy to check in practice. The test presented here can be easily implemented in practice, since it only depends on the *observed* sparse approximation to determine its optimality. When the test is satisfied, the ideal sparse approximation (best M -term approximation) can easily be recovered, because the found solution and the optimal have the same support.
- the test is independent of the particular algorithm used to get the sparse approximation: there is no need to make a new proof or find new optimality conditions when one introduces a new algorithm. Emphasis is indeed on the optimality *of a decomposition* rather than on the optimality *of an algorithm*, as in the work of Wohlberg [211].
- in the case where the error is measured with the mean square error (MSE) and the dictionary is incoherent, our test is close to being sharp (see Sections 5.7.4-5.7.5). Moreover, the test is satisfied in some cases where the residual seems “too large” for the previous contributions [48, 77, 91, 188, 189] to provide conclusive results. Indeed, one of the key contributions of this work is a new measure of the “size” of a residual which is less pessimistic than the worst-case measures based on the energy or the maximum correlation with the incoherent dictionary.
- besides the MSE, it is possible to deal with non-quadratic distortion measures, so one could imagine to insert visual criteria if one is dealing with images, or auditive criteria if one is dealing with sounds, or any other that is criteria more appropriate to the data or to the target application.
- not only the ℓ^0 and ℓ^1 sparsity measures are considered, but also all the ℓ^τ sparsity measures* $\|\cdot\|_\tau^\tau$, $0 \leq \tau \leq 1$, as well as a much larger class of “admissible” measures, as discussed in Section 5.3.

5.3 Main concepts and results

As in previous chapters, consider \mathcal{D} a dictionary of normalized atoms $\{\mathbf{g}_{\gamma_k}\}$, living in a finite or infinite dimensional real or complex vector space \mathcal{H} (which may be a Hilbert space or more generally

*Throughout this thesis, both $\|c\|_0^0$ and $\|c\|_0$ denote the ℓ^0 “norm”, which counts the number of nonzero coefficients in c .

a Banach space). Using various sparse approximation methods (Matching Pursuits [119, 143], Basis Pursuit [21], FOCUSS [80], etc.) one can decompose a signal $\mathbf{s} \in \mathcal{H}$ as

$$\mathbf{s} = \sum_k c_k \mathbf{g}_{\gamma_k} + \mathbf{e} \quad (5.1)$$

where the sequence $c = (c_k)$ is “sparse” and the residual \mathbf{e} is “small”. Throughout this chapter, Eq. (5.1) will be written $\mathbf{s} = \mathbf{\Phi}c + \mathbf{e}$, where $\mathbf{\Phi}$ corresponds to the synthesis operator which maps representation coefficients to signals for the dictionary \mathcal{D} . Thus, the representation coefficients c and the signal \mathbf{s} will be considered as column vectors, and the dictionary \mathcal{D} as a matrix $\mathbf{\Phi}$ of size $D \times N$ (with $N \geq D$), where D is the dimension of the signal space, and N is the number of elements in the dictionary. As in the previous chapters, bold characters denote signals (vectors in the space \mathcal{H}), and plain characters denote coefficient sequences.

Optimality of approximation Eq. (5.1) will be evaluated through two parameters: its goodness (how close is the approximated signal to the real one) and its sparsity. The goodness of approximation Eq. (5.1) can be measured by some distortion measure $d(\mathbf{e})$ (such as a norm on \mathcal{H}) which only depends on the residual \mathbf{e} . The sparsity of this representation c can be measured by an ℓ^τ norm ($0 \leq \tau \leq 1$) or more generally by an f -norm

$$\|c\|_f := \sum_k f(|c_k|), \quad (5.2)$$

where $f : [0, \infty) \rightarrow [0, \infty)$ is non-decreasing, not identically zero, and $f(0) = 0$. The smaller $\|c\|_f$, the sparser the representation c . The most popular sparsity measures are the ℓ^τ pseudo-norms $\|\cdot\|_\tau^\tau = \|\cdot\|_{f_\tau}$ where $f_\tau(t) := t^\tau$ for $0 \leq \tau \leq 1$ (with the convention $0^0 := 0$ and $t^0 = 1, t > 0$) but one can imagine many other more exotic sparsity measures, such as the ones presented in Appendix A.1. Of particular interest will be the class \mathcal{S} of **sub-additive** sparsity measures which, in addition to the above properties, satisfy

$$f(t+u) \leq f(t) + f(u) \text{ for all } t, u \geq 0,$$

and the class \mathcal{M} of **admissible** sparsity measures where

$$t \mapsto \frac{f(t)}{t} \text{ is non-increasing.}$$

It is easy to check that $\mathcal{M} \subset \mathcal{S}$, (see [87] and Appendix A.1). One can define a partial order [87] on \mathcal{S} by letting $f \ll g$ if, and only if, there is some $h \in \mathcal{M}$ such that $f = h \circ g$ (\mathcal{S} is stable by composition, see Appendix A.1). With respect to this partial order, the ℓ^0 and ℓ^1 “norms” are respectively the smallest and the largest admissible sparsity measures, in that $f_0 \ll f \ll f_1$ for each $f \in \mathcal{M}$ (where f_0 will be the ℓ^0 norm and f_1 , the ℓ^1).

Since different sparse approximation algorithms may optimize different sparsity criteria (ℓ^1 norm for Basis Pursuits, various ℓ^τ norms for FOCUSS, etc.), rely on various distortion measures, make a different compromise between sparsity and distortion, or even simply use a heuristic approach such as the greedy approach of Matching Pursuits, it is *a priori* hard to predict how solutions computed through different algorithms are related to one another. Our main theorems provide a simple test to check *a posteriori* if a computed decomposition $\mathbf{s} = \mathbf{\Phi}c + \mathbf{e}$ is nearly optimal, in the sense that c is close to any representation c' which is both sparser and leads to a smaller distortion (basically, they imply that c and c' , have the same support for all the components that are bigger than a threshold).

5.3.1 Main theorems in a Hilbert space

To state the theorems, few notations have to be introduced first. Let \mathcal{H} be a Hilbert space equipped with the norm $\|\mathbf{s}\|_{\mathcal{H}}^2 = \langle \mathbf{s}, \mathbf{s} \rangle$ where $\langle \cdot, \cdot \rangle$ denotes the inner product. For each integer K w,denote

$$\sigma_{\min,K}^2(\mathcal{D}) := \inf_{\|\delta\|_0 \leq K} \frac{\|\Phi\delta\|_{\mathcal{H}}^2}{\|\delta\|_2^2} \leq 1 \quad (5.3)$$

and consider the norm (not to be confused with the ℓ^K norm $\|\cdot\|_K$)

$$|\mathbf{e}|_K := \sqrt{\sum_{k \in I_K(\mathbf{e})} |\langle \mathbf{e}, \mathbf{g}_k \rangle|^2}, \quad (5.4)$$

where $I_K(\mathbf{e})$ indexes the K largest inner products $|\langle \mathbf{e}, \mathbf{g}_k \rangle|$. $|\mathbf{e}|_K$ is a measure of the correlation of the dictionary basis functions and the residual. This measure depends on the number of terms of the decomposition, on the residual and, even though it is not explicit in the notation, it is easy to see that it also depends on the dictionary \mathcal{D} . Since the dictionary is generally fixed, notations will often be simplified by omitting \mathcal{D} in some quantities that depend on it. In infinite dimension, $|\cdot|_K$ is generally not equivalent to the native norm $\|\cdot\|_{\mathcal{H}}$. However, for any integer K , it can easily be proved, by applying the norm definition, that

$$\sup_k |\langle \mathbf{e}, \mathbf{g}_{\gamma_k} \rangle| = |\mathbf{e}|_1 \leq |\mathbf{e}|_K \leq \sqrt{K} \cdot |\mathbf{e}|_1 \leq \sqrt{K} \cdot \|\mathbf{e}\|_{\mathcal{H}}, \quad (5.5)$$

so the norms $|\cdot|_K$ for different K are equivalent. The previous definitions very naturally lead to the following result:

Theorem 5.1 *Let c , such that $\mathbf{s} = \Phi c + \mathbf{e}$, be a sparse approximation of a signal \mathbf{s} , which may have been computed with any algorithm. Let $M := \|c\|_0^0$ and let c' be any other representation. If $\|\mathbf{s} - \Phi c'\|_{\mathcal{H}} \leq \|\mathbf{s} - \Phi c\|_{\mathcal{H}}$ and $\|c'\|_0^0 \leq \|c\|_0^0$, then*

$$\|c' - c\|_{\infty} \leq \frac{|\mathbf{e}|_1 + |\mathbf{e}|_{2M}}{\sigma_{\min,2M}^2} \quad (5.6)$$

$$\|c' - c\|_2 \leq \frac{2 \cdot |\mathbf{e}|_{2M}}{\sigma_{\min,2M}^2} \quad (5.7)$$

$$\|\Phi c' - \Phi c\|_{\mathcal{H}} \leq \frac{2 \cdot |\mathbf{e}|_{2M}}{\sqrt{\sigma_{\min,2M}^2}} \quad (5.8)$$

This theorem gives a criteria of optimality of a sparse signal approximation. Generally, the bound (5.8) is better than the “worst case” one $\|\Phi c' - \Phi c\|_{\mathcal{H}} = \|(\mathbf{s} - \Phi c') - (\mathbf{s} - \Phi c)\|_{\mathcal{H}} \leq 2\|\mathbf{e}\|_{\mathcal{H}}$. This point will be further commented in Section 5.6.

The second theorem presented here is the generalization of the first one. It is stronger, since it is valid for *any* admissible sparsity measure. A few additional definitions are needed to present the second theorem.

Let $\Phi_I : \ell^2(I) \rightarrow \mathcal{H}$ denote the synthesis matrix associated to the subdictionary $\{\mathbf{g}_k, k \in I\}$ and $\Phi_I^+ = (\Phi_I^* \Phi_I)^{-1} \Phi_I^*$ be its Moore-Penrose pseudo-inverse, where $(\cdot)^*$ denotes the adjoint (or in matrix terminology the complex conjugate transpose), that is to say $\Phi^* \mathbf{e} := (\langle \mathbf{e}, \mathbf{g}_{\gamma_k} \rangle)_k$. Then, much inspired by the Exact Recovery Coefficient introduced in [189], consider

$$\lambda_M(\Phi) := 1 - \sqrt{M} \cdot \sup_{\text{card}(I) \leq M} \sup_{k \notin I} \|\Phi_I^+ \mathbf{g}_{\gamma_k}\|_2. \quad (5.9)$$

Then, the generalization for any sparsity measure of Theorem 5.1 is as follows:

Theorem 5.2 *Let c , such that $\mathbf{s} = \Phi c + \mathbf{e}$, be a sparse approximation of a signal \mathbf{s} , which may have been computed with any algorithm. Let $M := \|\mathbf{x}\|_0^0$ and assume that $\lambda_M(\Phi) > 0$. Let c' be any other representation: if $\|\mathbf{s} - \Phi c'\|_{\mathcal{H}} \leq \|\mathbf{s} - \Phi c\|_{\mathcal{H}}$ and if there exists some admissible sparsity measure f such that $\|c'\|_f \leq \|c\|_f$, then*

$$\|c' - c\|_{\infty} \leq \frac{2}{\lambda_M^2} \cdot \frac{|\mathbf{e}|_1 + |\mathbf{e}|_M}{\sigma_{\min, M}^2} \quad (5.10)$$

$$\|c' - c\|_2 \leq \frac{4 \cdot \sqrt{1+M} \cdot |\mathbf{e}|_M}{\sigma_{\min, M}^2 \cdot \lambda_M^2} \quad (5.11)$$

$$\|\Phi c' - \Phi c\|_{\mathcal{H}} \leq \frac{4 \cdot |\mathbf{e}|_M}{\sqrt{\sigma_{\min, M}^2 \cdot \lambda_M^2}} \quad (5.12)$$

Note that, in Eqs. (5.10)-(5.12), $2M$ has been replaced with M in the subscripts for $|\mathbf{e}|_{(\cdot)}$ and $\sigma_{\min, (\cdot)}^2$ compared to Eqs. (5.6)-(5.8). The proof of the two previous theorems is not straightforward, and will be done through the following sections of this chapter.

Corollary 5.1 (Test of ℓ^0 optimality) *Under the hypotheses and notations of Theorem 5.1, assume that*

$$|\mathbf{e}|_1 + |\mathbf{e}|_{2M} < \frac{\sigma_{\min, 2M}^2}{2} \cdot \min_{\{k: |c_k| \neq 0\}} |c_k|. \quad (5.13)$$

If c' satisfies $\|\mathbf{s} - \Phi c'\|_{\mathcal{H}} \leq \|\mathbf{s} - \Phi c\|_{\mathcal{H}}$ and $\|c'\|_0^0 \leq \|c\|_0^0$, then c' and c have the same ‘‘support’’:

$$\text{sup}(c') := \{k : |c'_k| \neq 0\} = \{k : |c_k| \neq 0\} = \text{sup}(c)$$

and the same sign: $\text{sign}(c_k) = \text{sign}(c'_k)$, for all k .

Proof This corollary is very easy to prove. Combining Eq. (5.6) of Theorem 5.1 and Eq. (5.13), one gets:

$$2\|c' - c\|_{\infty} < \min_{\{k: |c_k| \neq 0\}} |c_k|, \quad (5.14)$$

which, as $\|c'\|_0^0 \leq \|c\|_0^0$, it is only possible if c and c' have the same support and the same sign. \square

Corollary 5.2 (Test of strong optimality) *Under the hypotheses and notations of Theorem 5.2, assume that*

$$|\mathbf{e}|_1 + |\mathbf{e}|_M < \frac{\sigma_{\min, M}^2 \cdot \lambda_M^2}{4} \cdot \min_{\{k: |c_k| \neq 0\}} |c_k|. \quad (5.15)$$

If c' satisfies $\|\mathbf{s} - \Phi c'\|_{\mathcal{H}} \leq \|\mathbf{s} - \Phi c\|_{\mathcal{H}}$ and if there exists some admissible sparsity measure f such that $\|c'\|_f \leq \|c\|_f$, then c' and c have essentially the same support:

$$\{k : |c'_k| > \theta\} = \text{support}(c), \text{ with } \theta := \frac{1}{2} \min_{\{k: |c_k| \neq 0\}} |c_k|.$$

Moreover for $k \in \text{support}(c)$ we have $\text{sign}(c_k) = \text{sign}(c'_k)$.

The proof for Corollary 5.2 is done similarly to the one of Corollary 5.1, but using Eq. (5.10) of Theorem 5.2 instead of Eq. (5.6) of Theorem 5.1.

Corollary 5.3 (Solution of the NP-hard problem) *If c and \mathbf{e} satisfy either the test (5.13) or the test (5.15), then the best M -term approximation Φc_M to $\mathbf{s} = \Phi c + \mathbf{e}$ is exactly the orthogonal projection of \mathbf{s} onto $\text{span}\{\mathbf{g}_{\gamma_k}, k \in \text{support}(c)\}$.*

This corollary is straightforward: once the set of atoms that form the optimal solution is known, it is obvious that the orthogonal projection of \mathbf{s} onto this set of atoms will be the one that will minimize the error.

It is worth remarking that the tests proposed in Corollaries 5.1-5.2 are reminiscent of some results of Tropp [189, Correlation Condition Lemma, Theorem 5.2], but with $\sup_k |\langle \mathbf{e}, \mathbf{g}_{\gamma_k} \rangle| = \|\mathbf{e}\|_1$ replaced with $\|\mathbf{e}\|_1 + \|\mathbf{e}\|_K$ for $K \in \{M, 2M\}$.

5.3.2 What do these results mean?

Chapter 2 presented several image models, and several algorithms that try to represent an image as a sum of basis functions with certain features that take into account some features of the model, such as the presence of regular functions that are edges. Chapter 4 presented several algorithms that consider the signal model to be just a sparse sum of terms plus a noise, and that try to solve this problem. It is maybe worth to stop a while here and further discuss about what a model, an algorithm and a problem are, and how do the results presented here relate to these three terms. So, what is a model, a problem and an algorithm?

- A model is a description of how a signal could have been generated, typically with a probabilistic prior in the Bayesian point of view or with parameters. For images, several models have been introduced in Chapter 2, that consider images as a superposition of objects with occlusions, or of geometrical features, or simply as a sum of basis functions. The two first image models are often used to give probabilistic prior to the third.
- A problem is an optimization problem, independently of how hard it is, what algorithm can solve it, etc. A problem can correspond to a model if, for example, it is the maximum likelihood (ML) or maximum a posteriori (MAP) estimation of model parameters, but it can also be difficult to make a model that fits a given problem.
- An algorithm is a function that takes an input and computes an output, a computer program, independently of which problem it can solve. It is just a tool that may, in some cases, solve a given problem according to a certain model.

Now, how can the results of this chapter give connections among models, problems and algorithms? Imagine that some *algorithm* has decomposed a signal \mathbf{s} as $\mathbf{s} = \Phi \mathbf{c} + \mathbf{e}$, where the number of terms that play a role in this decomposition $M := \|\mathbf{c}\|_0$ is small enough ($\lambda_M > 0$ and $\lambda_M^2 \sigma_{\min, M}^2 > 0$). If this is the case, and if the test gives a positive result (the residual is small enough compared to the value of the coefficients that have been found in \mathbf{c} , then the computed coefficients \mathbf{c} are ‘not too far’ from the solutions of each of the (generally non-convex) *optimization problems*

$$\min_{\mathbf{c}'} \|\mathbf{c}'\|_f \text{ subject to } \|\mathbf{s} - \Phi \mathbf{c}'\|_{\mathcal{H}} \leq \epsilon \quad (5.16)$$

with $\epsilon := \|\mathbf{s} - \Phi \mathbf{c}\|_{\mathcal{H}} = \|\mathbf{e}\|_{\mathcal{H}}$. Therefore:

- for the input signal \mathbf{s} , the *problems* (5.16) for different sparsity measures f (thus, different *models*) have solutions close to one another;
- on the input signal \mathbf{s} , the *algorithm* which produced the decomposition $\mathbf{s} = \Phi \mathbf{c} + \mathbf{e}$ nearly solved each of these *problems*.

Corollary 5.3 shows that if the residual is small enough, one can actually directly ‘jump’ from the computed coefficients \mathbf{c} to the solution \mathbf{c}_0 of the NP-hard best M -term approximation problem

$$\min_{\mathbf{c}'} \|\mathbf{s} - \Phi \mathbf{c}'\|_{\mathcal{H}} \text{ subject to } \|\mathbf{c}'\|_0 \leq M.$$

So, the results say that a given *algorithm* was able to solve an instance of a *problem*, that corresponds to several *models*. It has to be said, though, that it is only for this specific instance, and that the algorithm could fail for another instance.

Now, assume that an observed signal \mathbf{s} follows the sparse model $\mathbf{s} = \Phi\mathbf{c} + \mathbf{e}$ where (with high probability) $\|\mathbf{c}\|_0 \leq M$ and $\|\mathbf{e}\|_{\mathcal{H}} = \epsilon$. One can consider the problem of estimating the coefficients c or the signal $\Phi\mathbf{c}$, which is a classical denoising problem. Theorem 5.2 shows that (with high probability)

- one can robustly estimate c by solving any of the sparse approximation problems (5.16);
- in particular, one can robustly estimate c by solving the convex ℓ^1 -minimization problem

$$\min_{c'} \|\mathbf{c}'\|_1 \text{ subject to } \|\mathbf{s} - \Phi\mathbf{c}'\|_{\mathcal{H}} \leq \epsilon \quad (5.17)$$

which can be done using any Quadratic Programming algorithm.

Solving the problem (5.16) is equivalent to solving the Lagrangian problem

$$\min_{c'} \|\mathbf{s} - \Phi\mathbf{c}'\|_{\mathcal{H}}^2 + \alpha \|\mathbf{c}'\|_1 \quad (5.18)$$

for an appropriate Lagrange multiplier $\frac{1}{\alpha} > 0$. Thus, the solution of (5.17) is also the Maximum A Posteriori (MAP) estimate of c under a Laplacian model on c and a Gaussian model on \mathbf{e} and we conclude that replacing the original sparse model with a Laplacian+Gaussian model does not significantly change the value of the estimate, yet it simplifies a lot its computation.

In practice, just as in [48, 189] a crucial practical problem is to estimate the noise level ϵ , which is unknown, or equivalently to tune the Lagrange multiplier used in the Quadratic Programming algorithm that solves (5.18). Further work is needed to investigate what can be said about the accuracy of the estimate when the exact noise level is unknown. Nevertheless, with the results presented here, and if the dictionary in use is incoherent enough so that $\lambda_M^2 \sigma_{\min, M}^2 > 0$, one can find a solution with Eq. (5.18) for a big α . If the result found has a positive sparsity result in the test presented, here, α can be slightly decreased, and so on until the results are no more positive. Then, the maximum number of terms for which sparsity is present with probability one will have been found.

5.3.3 Explicit tests of optimality in a Hilbert space

To apply these tests in practice, explicit (lower) estimates of the numbers $\lambda_M(\Phi)$ and $\sigma_{\min, K}^2(\Phi)$, which –so far– may seem fairly abstract are needed. Indeed, these parameters are, again, NP-hard to compute, thus, some approximation is needed. For M sufficiently small, such estimates can be obtained using the Babel function $\mu_1(M, \Phi)$ (which is also quite computationally complex, but which bound can be more easily estimated [188]), already defined in the previous chapter and in [188, 189, 191], but reminded here

$$\mu_1(M, \mathcal{D}) := \sup_{\text{card}(I) \leq M} \sup_{k \notin I} \sum_{i \in I} |\langle \mathbf{g}_k, \mathbf{g}_i \rangle|, \quad (5.19)$$

as well as the 2-Babel function which we define as

$$\mu_2(M, \mathcal{D}) := \sup_{\text{card}(I) \leq M} \sup_{k \notin I} \sqrt{\sum_{i \in I} |\langle \mathbf{g}_k, \mathbf{g}_i \rangle|^2}. \quad (5.20)$$

Proposition 5.1 *Let \mathcal{D} be a normalized dictionary in a Hilbert space \mathcal{H} . If $\mu_1(2M - 1) < 1$ then*

$$\sigma_{\min, 2M}^2 \geq 1 - \mu_1(2M - 1) > 0 \quad (5.21)$$

If $\sqrt{M}\mu_2(M) + \mu_1(M - 1) < 1$ then $\lambda_M > 0$ and

$$\sigma_{\min, M}^2 \cdot \lambda_M^2 \geq \frac{\left(1 - \sqrt{M}\mu_2(M) - \mu_1(M - 1)\right)^2}{1 - \mu_1(M - 1)} \quad (5.22)$$

Proof: In order to keep the flow of the explanations, the proof of Proposition 5.1 is postponed to Section 5.7.3. \square

The test can be done by applying the limit for $\sigma_{\min, M}^2 \cdot \lambda_M^2$, given by Eq. (5.22), to Corollary 5.2. It has to be taken into account that if the test is positive, you are sure you have the sparsest solution. On the other hand, you may have a negative test and still have the sparsest solution. The test is, nevertheless, sharper than similar tests presented in previous works, and it has the advantage that it is algorithm independent.

5.3.4 Examples

Orthonormal basis

When \mathcal{D} is an orthonormal basis, we have $\mu_1(M) = 0$ and $\mu_2(M) = 0$ for all M , hence the test of ℓ^0 optimality takes the simple form

$$|\mathbf{e}|_1 + |\mathbf{e}|_{2M} < \min_{\{k: |c_k| \neq 0\}} \frac{|c_k|}{2}$$

which turns out to be sharp (see Section 5.5). The test of strong optimality becomes

$$|\mathbf{e}|_1 + |\mathbf{e}|_M < \min_{\{k: |c_k| \neq 0\}} \frac{|c_k|}{4}$$

and it is also sharp up to a constant factor (see Sections 5.7.4-5.7.5).

Union of incoherent orthonormal bases

When \mathcal{D} is a union of two or more maximally incoherent orthonormal bases in \mathbb{C}^N , such as the Dirac basis, the Fourier basis

$$\mathbf{g}_k[n] = e^{\frac{2\pi kn}{N}i}, \quad 0 \leq k \leq N - 1 \quad (5.23)$$

and the Chirp basis

$$\mathbf{g}_k[n] = e^{\frac{\pi kn^2}{N}i}, \quad 0 \leq k \leq N - 1, \quad (5.24)$$

$\mu_1(M) = \frac{M}{\sqrt{N}}$ and $\mu_2(M) = \sqrt{\frac{M}{N}}$. It follows that, for

$$M \leq \frac{(1 + \frac{\sqrt{N}}{3})}{2}, \quad (5.25)$$

$\sigma_{\min, 2M}^2 \geq \frac{2}{3}$ and $\sigma_{\min, M}^2 \cdot \lambda_M^2 \geq \frac{4}{9}$, so the conclusions of Corollary 5.1 (resp. Corollary 5.2) hold if

$$\begin{aligned} |\mathbf{e}|_1 + |\mathbf{e}|_{2M} &< \min_{\{k: |c_k| \neq 0\}} \frac{|c_k|}{3} \\ \text{or} \\ |\mathbf{e}|_1 + |\mathbf{e}|_M &< \min_{\{k: |c_k| \neq 0\}} \frac{|c_k|}{9}, \end{aligned}$$

respectively.

More generally, since in this case $\mu_1(2M - 1) = \sqrt{M}\mu_2(M) + \mu_1(M - 1) = \frac{(2M-1)}{\sqrt{N}}$, one can apply the tests whenever $M < \frac{(1+\sqrt{N})}{2}$. As M gets closer to $\frac{1+\sqrt{N}}{2}$, the bound on the allowed size of the residual decreases and the tests becomes more restrictive. For three maximally incoherent orthonormal bases in dimension $N = 1024$ and $M \leq 16 < \frac{33}{2}$, both tests can be applied to guarantee the optimality of a sparse approximation. In comparison, without such a test, one would have to compare the quality of the observed approximation with that of $\binom{M}{3N} - 1$ other M -term approximations.

A numerical example

A numerical example to illustrate how the test can be applied in practice may be helpful for the reader to fully understand it. To mimic “musical notes” and transients” of audio signal, consider a dictionary $\Phi = [C, E, G, B, I_{25}]$, which is the union of an orthonormal basis of deltas with a set of 5 (normalized) sinusoids, that is to say I_N is the identity matrix of dimension N and C, E, G, B are unit vectors proportional to the sinusoids

$$\sin\left(\frac{2\pi \cdot \text{freq} \cdot n}{2048}\right), \quad n = 1 \dots 25. \quad (5.26)$$

with respective frequencies $\text{freq} = 261, 330, 392$ and 494 . These frequencies correspond to the fundamental frequency of the notes C, E, G, B sampled at $2048Hz$ (this is obviously an unrealistic sampling frequency for audio signals but the example is rather a toy for illustration here). For this dictionary, it is easy to compute the first values of the Babel functions. In particular, we can compute $\mu_1(1) \approx 0.2831$ and $\mu_2(2) \approx 0.3998$ and using Eq. (5.22) we obtain that $\sigma_{\min,2}^2 \cdot \lambda_2^2 > 0.2113$.

Given a signal of 25 samples \mathbf{s} (which we generated as a superimposition of the ‘note’ C , with coefficient 15 and the ‘note’ E , with coefficient 10, together with some additive Gaussian white noise), we can use various sparse approximation algorithms to decompose it in the dictionary Φ . We performed $M = 2$ steps of Orthogonal Matching Pursuit (OMP) and found that $\mathbf{s} = \Phi\mathbf{c} + \mathbf{e}$, with

$$\mathbf{c} \approx [15.0080, 10.0276, 0, \dots, 0]^T \quad (5.27)$$

Using the correlations of the residual \mathbf{e} with the atoms of the dictionary (which were already computed as a natural by-product of the two steps of OMP) we computed $|\mathbf{e}|_1 \approx 0.0936$ and $|\mathbf{e}|_M = |\mathbf{e}|_2 \approx 0.1311$. Thus, we were able to check that

$$|\mathbf{e}|_1 + |\mathbf{e}|_M \approx 0.2247 < \frac{0.2213 \cdot 10.0276}{4}$$

which ensures that the strong optimality test is satisfied. Thus, we are sure that the two atoms found by OMP are exactly the two atoms of the best 2 term approximation to the signal. After an additional iteration of OMP, we obtained an $M = 3$ term approximation to the signal, and it turned out that the new residual and the coefficients no longer satisfied the test (because the smallest coefficient in \mathbf{c} was of the same order as the noise level). Thus, we were no longer sure that the three term expansion provided by three iterations of OMP was “optimal” and we stopped the iterations. However, in order to use this in more realistic examples, such as building a good stopping criterion for Matching Pursuit, it has to be taken into account that the fact that the four term test is negative it can mean two things: or that three is the maximum number that have to be taken into account in the sparse approximation or that the four term sparse approximation is not unique (but this last does not exclude the fact that the fifth term sparse approximation can give positive test results).

Some remarks about the test

It has to be noticed that the breakdown point of the test does not only depend on the number of terms and the coherence of the dictionary in use. Imagine a signal that has an sparse representation on an orthonormal basis as follows:

$$[A, B, B, C, D, E, 0, 0, 0, 0, 0, 0, 0],$$

with $|A| > |B| > |C| > |D| > |E|$. Imagine now that a two term decomposition has been done:

$$[A, B, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0].$$

For this solution, the test would be negative, and the reason for this is that the above solution, even though is the sparsest, is not unique. But now, if the three term decomposition is wished, and the solution found is:

$$[A, B, B, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],$$

if the values of the residual are small enough, the test would be positive for three terms. It is thus not possible to show the breakpoint of the test with respect to the number of coefficients and the redundancy of a given dictionary, given that this breakpoint is strongly signal dependent.

5.4 Core elements of the proofs

Now that the main results how they can be used have been explained, it is worth to introduce some technical definitions and lemmas which are at the core of the proof of our theorems. This section is stated in the most general setting, and how some quantities can be estimated in specific cases will be left for later.

Let c , with $\mathbf{s} = \Phi c + \mathbf{e}$, be a sparse approximation of a signal \mathbf{s} . Let $M := \|c\|_0^0$ and assume that, for a fixed f , c' satisfies $d(\mathbf{s} - \Phi c') \leq d(\mathbf{s} - \Phi c)$ and $\|c'\|_f \leq \|c\|_f$. Letting $\delta := c' - c$, $\delta \in D_d(\mathbf{e}, \Phi) \cap E_f(C_M)$ with

$$D_d(\mathbf{e}, \Phi) := \left\{ \delta : d(\mathbf{e} - \Phi \delta) \leq d(\mathbf{e}) \right\} \quad (5.28)$$

$$E_f(C) := \bigcup_{z \in C} \left\{ \delta : \|z + \delta\|_f \leq \|z\|_f \right\} \quad (5.29)$$

and $C_M := \{c : \|c\|_0^0 \leq M\}$. $D_d(\mathbf{e}, \Phi)$ is then the set of δ that, when applied to the dictionary synthesis matrix Φ and subtracted to the residual of a sparse decomposition, reduce the distortion of this decomposition, and $E_f(C)$ the set of δ that increase the sparsity of a sequence. Thus, for $0 < q < \infty$:

$$\|c' - c\|_q \leq \sup_{\delta \in D_d(\mathbf{e}, \Phi) \cap E_f(C_M)} \|\delta\|_q, \quad (5.30)$$

and

$$d(\Phi c' - \Phi c) \leq \sup_{\delta \in D_d(\mathbf{e}, \Phi) \cap E_f(C_M)} d(\Phi \delta). \quad (5.31)$$

In the following we will simply denote $D(\mathbf{e})$ since Φ and $d(\cdot)$ are generally fixed. The results of this paper follow from upper estimates of the suprema in Eqs. (5.30)-(5.31), using the following lemma:

Lemma 5.1 *If f is a sub-additive sparsity measure then*

$$E_f(C_M) = \left\{ \delta : \sum_{k \in I_M(\delta)} f(|\delta_k|) \geq \frac{\|\delta\|_f}{2} \right\} \quad (5.32)$$

where $I_M(\delta)$ is the set of the M largest components of $|\delta_k|$.

To keep the flow of the argument, the proof of this Lemma is postponed to Appendix A.2. By [87, Lemma 7], for any admissible sparsity measure $h \in \mathcal{M}$, any sequence $z = (z_k)$ and any integer M , it holds that

$$\frac{\sum_{k \in I_M(z)} h(|z_k|)}{\|z\|_h} \leq \frac{\sum_{k \in I_M(z)} |z_k|}{\|z\|_1}. \quad (5.33)$$

Thus, whenever $f \ll g$, $E_f(C_M) \subset E_g(C_M)$ holds, and it follows that

$$\sup_{\delta \in D(\mathbf{e}) \cap E_f(C_M)} \|\delta\|_q \leq \sup_{\delta \in D(\mathbf{e}) \cap E_g(C_M)} \|\delta\|_q \quad (5.34)$$

$$\sup_{\delta \in D(\mathbf{e}) \cap E_f(C_M)} d(\Phi\delta) \leq \sup_{\delta \in D(\mathbf{e}) \cap E_g(C_M)} d(\Phi\delta). \quad (5.35)$$

Since $f_0 \ll f \ll f_1$ for every admissible sparsity measure f , we will only estimate the suprema in Eqs. (5.30)-(5.31) for $f = f_0$ and $f = f_1$.

5.5 Estimates in the canonical basis

Estimating the right hand side suprema in Eq. (5.34) (for $q = \infty$ and $q = 2$) and Eq. (5.35) with $d(\cdot)$ the Hilbertian norm immediately yields Theorem 5.1 and Theorem 5.2. Since these estimates are a bit technical, it is easier to postpone them for a while and to begin with an estimate for the simple case where \mathcal{D} is the canonical basis in an ℓ^p sequence space. This estimate is both illustrative and technically useful, since it provides the basic tools to obtain the more general estimates for arbitrary dictionaries in Hilbert spaces.

Lemma 5.2 *Consider $\Phi = \mathbf{B}$ the canonical basis in ℓ^p , $0 < p \leq \infty$, $d(\cdot) = \|\cdot\|_p$ and let $K \geq 1$. When $0 < p < \infty$ we have for any $b = (b_k) \in \ell^p$:*

$$\sup_{\substack{\|\delta\|_0 \leq K \\ d(b - \mathbf{B}\delta) \leq d(b)}} \|\delta\|_\infty = \sup_k |b_k| + \left(\sum_{k \in I_K(b)} |e_k|^p \right)^{1/p} \quad (5.36)$$

where $I_K(b)$ indexes the K largest components $|b_k|$. Similarly when $p = \infty$ we have

$$\sup_{\substack{\|\delta\|_0 \leq K \\ d(b - \mathbf{B}\delta) \leq d(b)}} \|\delta\|_\infty = 2 \cdot \sup_k |b_k|. \quad (5.37)$$

Notice that in the above Lemma, and in all this section, b denotes both a signal and a sequence of coefficients, hence according to our convention it could be written either in bold or in plain letters. The choice of the plain notation seemed more natural, because when the lemma will be used later on, b will be rather considered as a sequence.

Since $\delta \in E_{f_0}(C_M)$ if, and only if, $\|\delta\|_0^0 \leq 2M^*$, the Lemma 5.2 provides the exact value of

$$\sup_{\delta \in D(\mathbf{e}) \cap E_{f_0}(C_M)} \|\delta\|_\infty \quad (5.38)$$

by letting $K = 2M$ in Eqs. (5.36)-(5.37). For $p = 2$, if $b_k = \langle \mathbf{e}, \mathbf{g}_k \rangle$ for some dictionary Φ and some residual \mathbf{e} in a Hilbert space, it is not difficult to check that the value is exactly $|\mathbf{e}|_1 + |\mathbf{e}|_{2M}$ (see Eq. (5.4)).

Proof of Lemma 5.2. We begin with the case $0 < p < \infty$. Let δ with $\|\delta\|_0^0 \leq K$ and consider $I = \text{support}(\delta)$. If $\|b - \mathbf{B}\delta\|_p \leq \|b\|_p$ we have $\sum_{k \in I} |b_k - \delta_k|^p \leq \sum_{k \in I} |b_k|^p$. Thus, for any $j \in I$, we have

$$|b_j - \delta_j| \leq \left(\sum_{k \in I} |b_k|^p \right)^{1/p}$$

which implies

$$|\delta_j| \leq |b_j| + \left(\sum_{k \in I} |b_k|^p \right)^{1/p} \leq \sup_k |b_k| + \left(\sum_{k \in I_K(b)} |b_k|^p \right)^{1/p}. \quad (5.39)$$

So, the left hand side in Eq. (5.36) is no larger than the right hand side. To get the converse inequality let $I = I_K(b)$ and $j \in I$ with $|b_j| = \sup_k |b_k|$, let $\delta_k := 0$, $k \notin I$, $\delta_k := b_k$, $k \in I, k \neq j$, and

$$\delta_j := \text{sign}(b_j) \cdot \left(|b_j| + \left(\sum_{k \in I} |b_k|^p \right)^{1/p} \right).$$

Obviously $\|\delta\|_0^0 \leq K$, $\|b - \mathbf{B}\delta\|_p \leq \|b\|_p$ and $\|\delta\|_\infty$ is no smaller than the right hand side in Eq. (5.36). The case $p = \infty$ is even easier: if $\|b - \mathbf{B}\delta\|_\infty \leq \|b\|_\infty$ then

$$\|\delta\|_\infty = \|b - \mathbf{B}\delta - b\|_\infty \leq \|b - \mathbf{B}\delta\|_\infty + \|b\|_\infty \leq 2\|b\|_\infty,$$

thus the left hand side in Eq. (5.37) is no larger than the right hand side. To get the converse inequality, assume for the sake of simplicity that there is an index l such that $|b_l| = \|b\|_\infty$. Letting $\delta_l = -2b_l$ and $\delta_k = 0$, $k \neq l$ we get a sequence $\delta = (\delta_k)$ which satisfies $\|b - \mathbf{B}\delta\|_\infty = \|b\|_\infty$ and $\|\delta\|_0^0 = 1 \leq K$. The same argument can be adapted to the case where the ℓ^∞ norm is not attained. \square

5.6 Sub-additive distortion measures

When \mathcal{D} is not the canonical basis or $d(\cdot)$ is not an ℓ^p norm, it is difficult to get exact estimates. Exact estimates are here substituted by “worst case” upper estimates for sub-additive distortion measure, *i.e.* when for any \mathbf{u} and \mathbf{v} we have $d(\mathbf{u} + \mathbf{v}) \leq d(\mathbf{u}) + d(\mathbf{v})$. This worst case estimates sill summarize the residual with its distortion, instead of using the full knowledge about it. No structure of the residual or the dictionary will be taken into account. Thus, the estimates will tell what happens with the “worst” residual which yields the same distortion. For the quadratic distortion measure, this approach will recover known results [48, 77, 189] on the identification of sparse approximations, since these were obtained using a worst case approach. However, we will see in the Section 5.7 how to prove Theorems 5.1-5.2 which provide much more precise bounds than the worst case ones.

*If $\|\delta\|_0^0 > 2M$, then it is impossible to have $\|z + \delta\|_f \leq \|z\|_f$

Lemma 5.3 *Let $d(\cdot)$ be a sub-additive distortion measure. Then*

$$\sup_{\delta \in D(\mathbf{e})} d(\Phi\delta) \leq 2 \cdot d(\mathbf{e}). \quad (5.40)$$

Proof. By definition, for any $\delta \in D(\mathbf{e})$, $d(\mathbf{e} - \Phi\delta) \leq d(\mathbf{e})$. Applying the sub-additivity of $d(\cdot)$, one gets

$$d(\Phi\delta) = d(\mathbf{e} - (\mathbf{e} - \Phi\delta)) \leq d(\mathbf{e}) + d(\mathbf{e} - \Phi\delta) \leq 2 \cdot d(\mathbf{e}).$$

□

Building upon this result, general (but somewhat abstract) upper estimates of the suprema in (5.30) are obtained. Denoting

$$A_q(C, \epsilon) := \sup_{\substack{z \in C \\ d(\Phi z) \leq \epsilon}} \|z\|_q, \quad \epsilon \geq 0, \quad (5.41)$$

Lemma 5.3 can be rewritten as:

$$\sup_{\delta \in D(\mathbf{e}) \cap E_f(C_M)} \|\delta\|_q \leq A_q(E_f(C_M), 2 \cdot d(\mathbf{e})). \quad (5.42)$$

This theoretically allows to express results similar to Theorem 5.1-5.2 for general sub-additive distortion measures. However, $A_q(E_f(C_M), \epsilon)$ are evidently not straightforward to compute. Even though estimating them for specific distortion measures (such as those designed to model auditive or visual distortion criteria) would be quite interesting, it is beyond the scope of this work. Instead, a more restricted case where more can be said will be introduced. If $d(\cdot)$ is not only sub-additive but it is indeed a norm, then $d(\lambda\mathbf{u}) = |\lambda| \cdot d(\mathbf{u})$ for any λ and \mathbf{u} , hence

$$A_q(E_f(C_M), \epsilon) = \epsilon \cdot A_q(E_f(C_M), 1) =: \epsilon \cdot A_q(E_f(C_M))$$

and it is possible to get an analogue to Theorem 5.1 and Corollary 5.1.

Theorem 5.3 *Let $d(\cdot)$ be a norm. Let c , such that $\mathbf{s} = \Phi c + \mathbf{e}$, be a sparse approximation of a signal \mathbf{s} , which may have been computed with any algorithm. Let $M := \|c\|_0^0$ and let c' be any other representation. If $d(\mathbf{s} - \Phi c') \leq d(\mathbf{e})$ and $\|c'\|_0 \leq \|c\|_0$, then*

$$d(\Phi c' - \Phi c) \leq 2 \cdot d(\mathbf{e}) \quad (5.43)$$

and for any $0 < q \leq \infty$

$$\|c' - c\|_q \leq 2 \cdot A_q(E_{f_0}(C_M)) \cdot d(\mathbf{e}). \quad (5.44)$$

Corollary 5.4 (Test of ℓ^0 optimality) *Under the hypotheses and notations of Theorem 5.3, assume that*

$$d(\mathbf{e}) < \frac{\min_{\{k: |c_k| \neq 0\}} |c_k|}{4 \cdot A_\infty(E_{f_0}(C_M))}. \quad (5.45)$$

If c' satisfies $d(\mathbf{s} - \Phi c') \leq d(\mathbf{s} - \Phi c)$ and $\|c'\|_0^0 \leq \|c\|_0^0$, then c' and c have the same support and sign.

This corollary is proved from Theorem 5.3 just as Corollary 5.1 is proved from Theorem 5.1. We let the reader express what would be the analogue of Theorem 5.2 and Corollaries 5.2-5.3. Even with $d(\cdot)$ a norm, computing $A_q(E_f(C_M))$ (or more realistically estimating it from above) seems difficult. The estimation with two particular norms of interest, the quadratic distortion and the maximum correlation with the dictionary vectors, will be introduced through two illustrative examples.

Example 5.1 When $d(\cdot) = \|\cdot\|_{\mathcal{H}}$ with \mathcal{H} a Hilbert space, the reader can easily check that for $q \geq 2$ one obtains

$$A_q(E_{f_0}(C_M)) \leq \sup_{\substack{\|\delta\|_0^2 \leq 2M \\ \|\Phi\delta\|_{\mathcal{H}} \leq 1}} \|\delta\|_2 = \frac{1}{\sqrt{\sigma_{\min, 2M}^2(\Phi)}}.$$

Combining the results obtained so far with Proposition 5.1, [48, Theorem 2.1] is recovered: if \mathbf{s} has a representation $\mathbf{s} = \Phi\mathbf{c} + \mathbf{e}$ with $M := \|\mathbf{c}\|_0^2 < \frac{1}{2} \left(1 + \frac{1}{\mu_1(1)}\right)$, and $\|\mathbf{e}\|_{\mathcal{H}} = \epsilon$, then the solution c_0 of the optimization problem

$$\min_{c'} \|c'\|_0^2 \quad \text{subject to} \quad \|\mathbf{s} - \Phi c'\|_{\mathcal{H}} \leq \epsilon \quad (5.46)$$

satisfies

$$\|c_0 - c\|_2^2 \leq \frac{4\epsilon^2}{1 - \mu_1(2M - 1)} \leq \frac{4\epsilon^2}{1 - \mu_1(1) \cdot (2M - 1)} \quad (5.47)$$

Notice that, compared to Theorem 5.1, $\|c_0 - c\|_2$ is upper estimated by $\|\mathbf{e}\|_{\mathcal{H}}$ instead of $2 \cdot |\mathbf{e}|_{2M}$. It will be seen in Sections 5.7.4-5.7.5 that the estimate presented here can give a much smaller bound than the worst case estimate. A similar analysis with an estimate of $A_q(E_{f_1}(C_M))$ would recover a result similar to [48, Theorem 3.1].

Example 5.2 In a Hilbert space \mathcal{H} , when $d(\cdot) = \sup_k |\langle \cdot, \mathbf{g}_k \rangle| = \|\Phi^* \cdot\|_{\infty}$, for $2 \leq q \leq \infty$, $A_q(E_{f_0}(C_M))$ can be upper-bounded by:

$$A_q(E_{f_0}(C_M)) \leq \sup_{\substack{\|\delta\|_0^2 \leq 2M \\ d(\Phi\delta) \leq 1}} \|\delta\|_2 \leq \frac{\sqrt{2M}}{\sigma_{\min, 2M}^2(\Phi)}$$

which is sharp for $q = 2$ when

mathcal{D} is a basis. The estimate is proved as follows: for any δ with $\|\delta\|_0^2 \leq 2M$, one obtains

$$\sigma_{\min, 2M}^2 \cdot \|\delta\|_2^2 \leq \|\Phi\delta\|_{\mathcal{H}}^2 = \langle \delta, \Phi^* \Phi \delta \rangle \leq \|\delta\|_1 \cdot d(\Phi\delta),$$

hence

$$\|\delta\|_2 \leq \frac{d(\Phi\delta)}{\sigma_{\min, 2M}^2} \cdot \frac{\|\delta\|_1}{\|\delta\|_2} \leq \frac{d(\Phi\delta)}{\sigma_{\min, 2M}^2} \cdot \sqrt{2M}.$$

As a result, if \mathbf{s} has a representation $\mathbf{s} = \Phi\mathbf{c} + \mathbf{e}$ with $M := \|\mathbf{c}\|_0^2 < \frac{1}{2} \left(1 + \frac{1}{\mu_1(1)}\right)$, and $\sup_k |\langle \mathbf{e}, \mathbf{g}_k \rangle| = \epsilon$, then the solution c_0 of the optimization problem

$$\min_{c'} \|c'\|_0^2 \quad \text{subject to} \quad \sup_k |\langle \mathbf{s} - \Phi c', \mathbf{g}_k \rangle| \leq \epsilon \quad (5.48)$$

satisfies

$$\|c_0 - c\|_2^2 \leq \frac{8M\epsilon^2}{1 - \mu_1(2M - 1)}. \quad (5.49)$$

5.7 Proof of the main results

When the distortion $d(\cdot)$ is the MSE, the worst case analysis carried out with general sub-additive measures can be drastically improved. The key observation is that

$$\delta \in D(\mathbf{e}) \iff |\langle \Phi^* \mathbf{e}, \delta \rangle| \geq \frac{\|\Phi\delta\|_{\mathcal{H}}^2}{2}. \quad (5.50)$$

Thanks to this, some problems can be recasted to a ℓ^2 sparse approximation problem in the canonical basis, for which we can use the results of Section 5.5. As a result, the obtained bounds will be close to sharp, as discussed below in Sections 5.7.4-5.7.5.

5.7.1 Proof of Theorem 5.1.

In all the following, it is assumed that $\delta \in D(\mathbf{e}) \cap E_{f_0}(C_M)$, that is to say $\|\delta\|_0^0 \leq 2M$ and $|\langle \Phi^* \mathbf{e}, \delta \rangle| \geq \|\Phi \delta\|_{\mathcal{H}}^2/2$. Denoting $\delta' = \sigma_{\min, 2M}^2(\Phi) \cdot \delta$, we have $\delta' \in E_{f_0}(C_M)$. The procedure to follow will be to check that

$$\|b - \mathbf{B}\delta'\|_2 \leq \|b\|_2 \text{ with } \mathbf{B} \text{ the canonical basis in } \ell^2 \text{ and}$$

b the restriction of $\Phi^* \mathbf{e}$ to the finite support of δ' , which is of size at most $2M$. Thus, Lemma 5.2 for $p = 2$ and $K = 2M$ will give exactly that

$$\sigma_{\min, 2M}^2(\Phi) \cdot \|\delta\|_{\infty} = \|\delta'\|_{\infty} \leq |\mathbf{e}|_1 + |\mathbf{e}|_{2M}$$

which will bring to the first inequality (5.6).

Indeed, using the assumptions on δ and the definition of $\sigma_{\min, 2M}^2(\Phi)$, one obtains:

$$\frac{|\langle b, \delta' \rangle|}{\|\delta'\|_2^2} = \frac{|\langle b, \delta \rangle|}{\sigma_{\min, 2M}^2(\Phi) \cdot \|\delta\|_2^2} = \frac{|\langle \Phi^* \mathbf{e}, \delta \rangle|}{\|\Phi \delta\|_{\mathcal{H}}^2} \cdot \frac{\|\Phi \delta\|_{\mathcal{H}}^2}{\sigma_{\min, 2M}^2(\Phi) \cdot \|\delta\|_2^2} \geq \frac{1}{2}.$$

It follows that $\|b - \mathbf{B}\delta'\|_2 \leq \|b\|_2$ as claimed. To get the second inequality (5.7), simply notice that

$$\|\delta\|_2^2 \leq \frac{2 \cdot |\langle b, \delta \rangle|}{\sigma_{\min, 2M}^2(\Phi)} \leq \frac{2 \cdot |\mathbf{e}|_{2M} \cdot \|\delta\|_2}{\sigma_{\min, 2M}^2(\Phi)},$$

where the above expression is obtained using the Cauchy-Schwarz inequality and the fact that $\|b\|_2 = |\mathbf{e}|_{2M}$. To get the third inequality (5.8) we observe that

$$\|\Phi \delta\|_{\mathcal{H}}^2 \leq 2 \cdot |\langle \Phi^* \mathbf{e}, \delta \rangle| \leq 2 \cdot |\mathbf{e}|_{2M} \cdot \|\delta\|_2 \leq \frac{4 \cdot |\mathbf{e}|_{2M}^2}{\sigma_{\min, 2M}^2(\Phi)}$$

where (5.7) has been used for achieving the right hand side inequality.

5.7.2 Proof of Theorem 5.2.

In all the following, it is assumed that $\delta \in D(\mathbf{e}) \cap E_{f_1}(C_M)$. Let $(|\langle \mathbf{e}, \mathbf{g}_{k_m} \rangle|)_{m \geq 1}$ and $(|\delta_{l_m}|)_{m \geq 1}$ be decreasing rearrangements of $(|\langle \mathbf{e}, \mathbf{g}_k \rangle|)_k$ and $(|\delta_k|)_k$. Let $c := \sigma_{\min, M}^2(\Phi) \cdot \lambda_M^2(\Phi)$, and $\delta' = (\delta'_m)$ by $\delta'_m = c \cdot |\delta_{l_m}|$ for $1 \leq m \leq M$ and $\delta'_m = 0$ for $m > M$. Similarly, let $b = (b_m)$ with $b_m = 2 \cdot |\langle \mathbf{e}, \mathbf{g}_{k_m} \rangle|$ for $1 \leq m \leq M$ and $b_m = 0$ for $m > M$.

Since $\|\delta'\|_0^0 \leq M$ and $\frac{|\langle \Phi^* \mathbf{e}, \delta \rangle|}{\|\Phi \delta\|_{\mathcal{H}}^2} \geq \frac{1}{2}$, if it is possible to prove that

$$\frac{|\langle b, \delta' \rangle|}{\|\delta'\|_2^2} \geq \frac{|\langle \Phi^* \mathbf{e}, \delta \rangle|}{\|\Phi \delta\|_{\mathcal{H}}^2} \tag{5.51}$$

then $\|b - \mathbf{B}\delta'\|_2 \leq \|b\|_2$ will be obtained, and Lemma 5.2 with $p = 2$ for $K = M$ will exactly give

$$c \cdot \|\delta\|_{\infty} = \|\delta'\|_{\infty} \leq 2 \cdot (|\mathbf{e}|_1 + |\mathbf{e}|_M)$$

which will bring to inequality (5.10).

(5.51) is proved by getting an upper bound on the numerator and a lower bound on the denom-

inator of the right hand side. For the numerator, the upper-bound is obtained as follows:

$$\begin{aligned}
|\langle \Phi^* \mathbf{e}, \delta \rangle| &\leq \sum_{m=1}^M |\langle \mathbf{e}, \mathbf{g}_{k_m} \rangle| \cdot |\delta_{l_m}| + \sum_{m \geq M+1} |\langle \mathbf{e}, \mathbf{g}_{k_m} \rangle| \cdot |\delta_{l_m}| \\
&\leq \sum_{m=1}^M |\langle \mathbf{e}, \mathbf{g}_{k_m} \rangle| \cdot |\delta_{l_m}| + |\langle \mathbf{e}, \mathbf{g}_{k_{M+1}} \rangle| \cdot \sum_{m \geq M+1} |\delta_{l_m}| \\
&\leq \sum_{m=1}^M |\langle \mathbf{e}, \mathbf{g}_{k_m} \rangle| \cdot |\delta_{l_m}| + |\langle \mathbf{e}, \mathbf{g}_{k_{M+1}} \rangle| \cdot \sum_{m=1}^M |\delta_{l_m}| \\
&= \sum_{m=1}^M (|\langle \mathbf{e}, \mathbf{g}_{k_m} \rangle| + |\langle \mathbf{e}, \mathbf{g}_{k_{M+1}} \rangle|) \cdot |\delta_{l_m}| \\
&\leq |\langle b, \delta' / c \rangle|
\end{aligned} \tag{5.52}$$

Now, it only remains to estimate the lower-bound for the denominator. Let $I = I_M(\delta)$ and consider δ_I the sequence with zeros everywhere except on I where it coincides with δ . Since $\Phi_I \Phi_I^+$ is an orthonormal projector and I is of cardinal M , it follows that

$$\|\Phi \delta\|_{\mathcal{H}}^2 \geq \|\Phi_I \Phi_I^+ \Phi \delta\|_{\mathcal{H}}^2 = \|\Phi_I z\|_{\mathcal{H}}^2 \geq \sigma_{\min, M}^2(\Phi) \cdot \|z\|_2^2 \tag{5.53}$$

with $z := \Phi_I^+ \Phi \delta = \delta_I + \sum_{k \notin I} \delta_k \cdot \Phi_I^+ \mathbf{g}_k$. So,

$$\begin{aligned}
\|z\|_2 &\geq \|\delta_I\|_2 - \sum_{k \notin I} |\delta_k| \cdot \sup_{k \notin I} \|\Phi_I^+ \mathbf{g}_k\|_2 \\
&\geq \|\delta_I\|_2 - \|\delta_I\|_1 \cdot \sup_{k \notin I} \|\Phi_I^+ \mathbf{g}_k\|_2 \\
&\geq \|\delta_I\|_2 - \sqrt{M} \|\delta_I\|_2 \cdot \sup_{k \notin I} \|\Phi_I^+ \mathbf{g}_k\|_2 \\
&\geq \lambda_M(\Phi) \cdot \|\delta_I\|_2
\end{aligned}$$

where the fact that $\delta \in E_{f_1}(C_M)$ is used in order to get the second inequality. If $\lambda_M(\Phi) > 0$, then

$$\|\Phi \delta\|_{\mathcal{H}}^2 \geq \sigma_{\min, M}^2(\Phi) \cdot \lambda_M^2(\Phi) \cdot \|\delta_I\|_2^2 = \frac{\|\delta'\|_2^2}{c}. \tag{5.54}$$

The desired inequality is obtained by combining (5.52) and (5.54)

$$\frac{|\langle \Phi^* \mathbf{e}, \delta \rangle|}{\|\Phi \delta\|_{\mathcal{H}}^2} \leq \frac{|\langle b, \delta' / c \rangle|}{\|\delta'\|_2^2 / c} = \frac{|\langle b, \delta' \rangle|}{\|\delta'\|_2^2}$$

The third inequality (5.12) is obtained by writing

$$\begin{aligned}
\|\Phi \delta\|_{\mathcal{H}}^2 &\leq 2 \cdot |\langle \Phi^* \mathbf{e}, \delta \rangle| \\
&\leq 2 \cdot \left| \left\langle b, \frac{\delta'}{c} \right\rangle \right| \\
&\leq 4 \cdot |\mathbf{e}|_M \cdot \|\delta_I\|_2 \\
&\leq 4 \cdot |\mathbf{e}|_M \cdot \frac{\|\Phi \delta\|_{\mathcal{H}}}{\sqrt{c}}
\end{aligned}$$

To get the second inequality (5.11), noticing that (5.54) and (5.12) yield

$$\|\delta_I\|_2^2 \leq \frac{\|\Phi \delta\|_{\mathcal{H}}^2}{c} \leq 4 \cdot \frac{|\mathbf{e}|_M \cdot \|\delta_I\|_2}{c},$$

which brings to $\|\delta_I\|_2 \leq 4 \cdot |\mathbf{e}|_M/c$, and using the fact that

$$\begin{aligned} \|\delta\|_2^2 &= \|\delta_I\|_2^2 + \sum_{k \notin I} |\delta_k|^2 \leq \|\delta_I\|_2^2 + \left(\sum_{k \notin I} |\delta_k| \right)^2 \\ &\leq \|\delta_I\|_2^2 + \|\delta_I\|_1^2 \leq (1+M) \cdot \|\delta_I\|_2^2. \end{aligned}$$

the inequality is achieved. \square

Notice that in the proof, instead of

$$b_m = 2 \cdot |\langle \mathbf{e}, \mathbf{g}_{k_m} \rangle|, \quad (5.55)$$

this sharper estimate could have been used instead:

$$b'_m = \frac{(|\langle \mathbf{e}, \mathbf{g}_{k_m} \rangle| + |\langle \mathbf{e}, \mathbf{g}_{k_{m+1}} \rangle|)}{2}. \quad (5.56)$$

This would have lead to slightly sharper estimates, however they would have been more cumbersome to express and, as discussed below in Sections 5.7.4-5.7.5, the simplified estimate with $b_m = 2 \cdot |\langle \mathbf{e}, \mathbf{g}_{k_m} \rangle|$ is almost sharp.

5.7.3 Proof of Proposition 5.1

Proposition 5.1 immediately follows from the lemmas below.

Lemma 5.4 *Let Φ be a normalized dictionary in a Hilbert space \mathcal{H} . Then, for every integer K*

$$\sigma_{\min, K}^2(\Phi) \geq 1 - \mu_1(K-1, \Phi). \quad (5.57)$$

The proof is based on Geršgorin Disk Theorem and can be found in [189].

Lemma 5.5 *Let Φ be a normalized dictionary in a Hilbert space \mathcal{H} . Then*

$$\lambda_M(\Phi) \geq 1 - \frac{\sqrt{M} \cdot \mu_2(M, \Phi)}{1 - \mu_1(M-1, \Phi)}. \quad (5.58)$$

Proof. For any index set I with $\text{card}(I) \leq M$:

$$\|(\Phi_I^* \Phi_I)^{-1}\|_{2,2} \leq \frac{1}{\sigma_{\min, M}^2(\Phi)} \leq \frac{1}{1 - \mu_1(M-1)}.$$

Since $\Phi_I^+ = (\Phi_I^* \Phi_I)^{-1} \Phi_I^*$ and, for $k \notin I$,

$$\|\Phi_I^* \mathbf{g}_k\|_2 = \sqrt{\sum_{i \in I} |\langle \mathbf{g}_k, \mathbf{g}_i \rangle|^2} \leq \mu_2(M),$$

which brings to

$$\|\Phi_I^+ \mathbf{g}_k\|_2 = \|(\Phi_I^* \Phi_I)^{-1} \Phi_I^* \mathbf{g}_k\|_2 \leq \frac{\mu_2(M)}{1 - \mu_1(M-1)}.$$

And the final result is achieved by using the definition of $\lambda_M(\Phi)$ (see Eq. (5.9)). \square

5.7.4 Sharpness of Theorem 5.1

When \mathcal{D} is an orthonormal basis, the estimate (5.6) in Theorem 5.1 is sharp since $\sigma_{\min, 2M}^2 = 1$ and we have an exact estimate given by Lemma 5.2. Similarly, one can check the sharpness of the estimates (5.7)-(5.8). For a general \mathcal{D} , a slight modification of the proof of Theorem 5.1 leads to

$$\sup_{\delta \in D(\mathbf{e}) \cap E_{f_0}(C_M)} \|\delta\|_\infty \geq \frac{|\mathbf{e}|_1 + |\mathbf{e}|_{2M}}{\sigma_{\max, 2M}^2} \quad (5.59)$$

with

$$\sigma_{\max, K}^2(\Phi) := \sup_{\|\delta\|_0 \leq K} \frac{\|\Phi\delta\|_{\mathcal{H}}^2}{\|\delta\|_2^2} \leq 1 + \mu_1(K-1).$$

Thus, if $\mu_1(2M-1)$ is small enough, the estimate (5.6) is almost sharp in the sense that it cannot be significantly improved.

5.7.5 Sharpness of Theorem 5.2

Since

$$\sup_{\delta \in D(\mathbf{e}) \cap E_{f_0}(C_M)} \|\delta\|_\infty \leq \sup_{\delta \in D(\mathbf{e}) \cap E_{f_1}(C_M)} \|\delta\|_\infty,$$

the (almost) sharpness of the results in Theorem 5.2 is a consequence of that of Theorem 5.1: combining Eq. (5.59) with the estimate (5.10), one obtains

$$\frac{|\mathbf{e}|_1 + |\mathbf{e}|_{2M}}{\sigma_{\max, 2M}^2} \leq \sup_{\delta \in D(\mathbf{e}) \cap E_{f_1}(C_M)} \|\delta\|_\infty \leq \frac{2 \cdot (|\mathbf{e}|_1 + |\mathbf{e}|_M)}{\sigma_{\min, M}^2 \cdot \lambda_M^2}.$$

When $\mu_1(2M-1)$ is small enough, the upper bounds become approximately $|\mathbf{e}|_1 + |\mathbf{e}|_{2M}$ and $2 \cdot (|\mathbf{e}|_1 + |\mathbf{e}|_M)$ which differ at most by a factor two since $|\mathbf{e}|_M^2 \leq |\mathbf{e}|_{2M}^2 \leq 2|\mathbf{e}|_M^2$.

5.8 Conclusions

Chapter 4 introduced several algorithms to obtain sparse approximations of a signal, and its capability to recover this sparse approximation. The natural evolution is to have a tool that allows to check whether a given sparse approximation (computed using any algorithm) is nearly optimal. This chapter has presented such a tool. Having a nearly optimal sparse approximation means that no other significantly different representation can, at the same time, be sparser and provide a better approximation for this input signal. The test is easy to implement, and it does not depend on the algorithm used to find the sparse signal approximation, neither does it rely on any prior knowledge of the ideal sparse approximation. In addition, the test can be used for several sparsity measures, and some extended results allow the use of non-quadratic distortion measures, which would allow to use it for more complex problems such as Bayesian estimation and signal denoising, with a fairly large class of sparse priors and random noise.

The results presented here could, maybe, be adapted to obtain optimality results on the optimality of simultaneous sparse approximation of several signals, so that they can be applied to blind source separation (or denoising, which, finally, is a blind source separation problem as well). The fact of being able to use several sparsity measures allows to use these results for the case where there is an *a priori* knowledge about the basis function distribution, such as the work presented in [39].

Finally, it has to be said that, even though this test is tighter than existing tests, still, the fact of using very redundant dictionaries and natural signals in most of the sparse approximation problems, causes that the test would give a negative result after a certain small number of decomposition terms.

Nevertheless, these results give hope toward being able to perform sparse signal approximations that approach the optimal solution.

Image approximation with Matching Pursuit

From all the techniques that allow to perform a signal approximation through a redundant dictionary, one has to choose the most appropriate for the data and the dictionary in use. Chapter 4 has enumerated some of these techniques, and presented in which situations one can be sure that a given technique has been able to find the sparsest possible approximation in a certain redundant dictionary. Among all these, Matching Pursuit (MP) seems the most appropriate for image approximation, due to several reasons: it guarantees that the residual energy decreases sufficiently fast when a coefficient is added, it allows a big flexibility in the dictionary of basis functions in use, and it is easy to implement. In addition, as it is an iterative method, it is less complex than other methods, such as BP, when having huge amounts of data, as it is the case with images and redundant dictionaries. Finally, MP can easily be parallelized. This chapter presents the implementation of an image approximation scheme through a dictionary formed by anisotropic and rotated basis functions through a MP algorithm. Several variations of this scheme are introduced here.

This chapter is structured as follows: Section 6.1 introduces the image model in use. Section 6.2 presents the kind of basis functions that are present in the dictionary in use, and the reasons of their choice. Section 6.3 presents a sub-optimal MP algorithm [65, 66], that uses a genetic algorithm to find the basis function at every iteration. Section 6.4 explains how a full search MP that is reasonably quick can be implemented [62], and compares some image decomposition results obtained with the genetic algorithm and with the FS-MP. Section 6.5 explains some tricks to optimize this FS-MP, specially in terms of memory usage. Section 6.6 discusses the flexibility of the presented MP image approximation scheme, and the problem of having a different scheme for the low-pass frequency band and the rest of the image. To overcome these limitations, Section 6.7 introduces a new dictionary of basis functions that gives a full image approximation with MP, without having to perform different coding schemes for the low and high frequencies. Section 6.8 introduces a weighted MP algorithm, that instead of taking energy as criteria for choosing the basis functions at every iteration, takes a weighted energy criteria. The weight can depend on several factors, and is based in an *a priori* knowledge that one may have of the image. In this case, the algorithm uses a dictionary formed by

three different subdictionaries, one for smooth regions, one for edge regions and one for textured regions. A previous image segmentation between smooth, edge and textured regions is then used as helping criteria for MP to place a function from the subdictionary of textures in the textured region, for example. The MP algorithms presented in Section 6.3, Section 6.4 and Section 6.8 are for gray-scale images. Section 6.9 explains how the MP algorithm can be used for color images, keeping the sparsity of the representation and profiting from the correlation among channels. Finally, some conclusions are given in Section 6.10.

6.1 Image Model

In order to efficiently represent a signal, there is a need to know some features of this signal and to make some assumptions. The main assumption that it is done with images is that they can be represented (or at least approximated with a residual that decreases when the number of terms increases) as a finite sum of M basis functions, with M smaller than the dimensionality of the signal (sparse representation, Section 2.6):

$$\mathbf{s} = \sum_{m=0}^{M-1} c_m \mathbf{g}_{\gamma_m}. \quad (6.1)$$

In order to have efficient representations, basis functions have to be adapted to the specific features of natural images, i.e. to contours. These are assumed to be continuous smooth functions, as done by several authors [42, 46, 64, 107]. It has been seen in Chapter 3 that variable segment length, a variable rotation and translation of the basis functions are necessary to have sparse image approximations. In addition, it is logical to think that an image is dilation, translation and rotation invariant, because in natural images, objects and contours can appear in any orientation or size or position. The dictionary of basis functions has to be designed according to all these image qualities. This implies that the dictionary has to be composed by a set of anisotropic scaling, rotation and translation of the mother basis function. In addition, the basis functions are supposed to be dilation, rotation and translation invariant, so that they match our model.

6.2 Dictionary of basis functions

6.2.1 Generating functions

The dictionary used in the scope of this thesis is built by varying the parameters of a basis function, in such a way that it generates an overcomplete set of functions spanning the input image space. The choice of the generating function, g , is driven by the idea of efficiently approximating contour-like singularities in 2-D. To achieve this goal, the atom is a smooth low resolution function in the direction of the contour, and behaves like a wavelet in the orthogonal (singular) direction. In other words, the dictionary is composed of atoms that are built on Gaussian functions along one direction, and on second derivative of Gaussian functions in the orthogonal direction:

$$\mathbf{g}(x, y) = \frac{2}{\sqrt{3\pi}} (4x^2 - 2) e^{-(x^2+y^2)}, \quad (6.2)$$

where $[x, y]$ is the vector of discrete image coordinates, and $\|g\| = 1$. The choice of the Gaussian envelope is motivated by the optimal joint spatial and frequency localization of this kernel. The second derivative occurring in the oscillatory component is a trade-off between the number of vanishing moments used to filter out smooth polynomial parts and ringing-like artifacts that may occur after strong quantization. It is also motivated by the presence of second derivative-like filtering in the early stages of the human visual system (see Chapter 2 and [124, 138, 139]).

6.2.2 Anisotropy and orientation

Anisotropic refinement and orientation is obtained by applying meaningful geometric transformations to the generating functions of unit L^2 norm, g , described here-above. These transformations can be represented by a family of unitary operators $U(\gamma)$, and the dictionary is thus expressed as:

$$\mathcal{D} = \{U(\gamma)g, \gamma \in \Gamma\}, \quad (6.3)$$

for a given set of indexes Γ . Basically this set must contain three types of operations:

- Translations \mathbf{b} , to move the atom all over the image.
- Rotations θ , to locally orient the atom along contours.
- Anisotropic scaling $\mathbf{a} = (a_x, a_y)$, to adapt to contour smoothness.

A possible action of $U(\gamma)$ on the generating atom \mathbf{g} is thus given by:

$$U(\gamma)\mathbf{g} = \mathcal{U}(\mathbf{b}, \theta)D(a_x, a_y)g \quad (6.4)$$

where \mathcal{U} is a representation of the Euclidean group,

$$\mathcal{U}(\mathbf{b}, \theta)g(\mathbf{p}) = g(\mathbf{r}_{-\theta}(\mathbf{p} - \mathbf{b})), \quad (6.5)$$

\mathbf{r}_{θ} is a rotation matrix, and D acts as an anisotropic dilation operator:

$$D(a_x, a_y)g(\mathbf{p}) = \frac{1}{\sqrt{a_x a_y}}g\left(\frac{x}{a_x}, \frac{y}{a_y}\right). \quad (6.6)$$

It is easy to prove that such a dictionary is overcomplete using the fact that, under the restrictive condition $a_x = a_y$, one gets 2-D continuous wavelets as defined in [4]. It is also worth stressing that, avoiding rotations, the parameter space is a group studied by Bernier and Taylor [8]. The advantage of such a parametrization is that the full dictionary is invariant under translation, rotation, and isotropic scaling (i.e. $a_x = a_y$), as desired in the specifications defined in Section 6.1.

6.2.3 Dictionary

Since the structured dictionary is built by applying geometric transformations to a generating mother function g , the atoms are therefore indexed by a string γ composed of five parameters: translation $\mathbf{b} = [b_x, b_y]$, anisotropic scaling $\mathbf{a} = [a_x, a_y]$ (with $a_x \leq a_y$ in order to have functions elongated on the direction of the contour) and rotation θ . Any atom in this dictionary can finally be expressed in the following form:

$$g_{\gamma}(x, y) = \frac{2}{\sqrt{3\pi}}(4g_x^2 - 2)\exp(-(g_x^2 + g_y^2)), \quad (6.7)$$

with

$$g_x = \frac{\cos(\theta)(x - b_x) + \sin(\theta)(y - b_y)}{a_x}, \quad (6.8)$$

and

$$g_y = \frac{\cos(\theta)(y - b_y) - \sin(\theta)(x - b_x)}{a_y}. \quad (6.9)$$

For practical implementations, all parameters in the dictionary must be discretized. For the Anisotropic Refinement (AR) Atoms sub-dictionary, the translation parameters can take any positive integer value smaller than the image dimensions (thus, the minimum translation allowed is of one

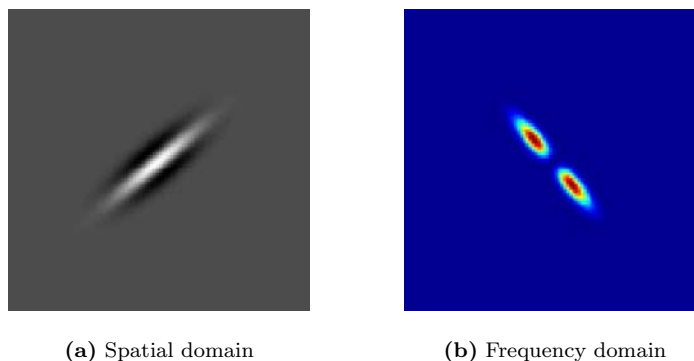


Figure 6.1: An example of one atom of the dictionary in the spatial domain and in the frequency domain.

pixel). The rotation parameter varies by increments of $\frac{\pi}{18}$, to ensure the over-completeness of the dictionary. If one wishes more approximation properties, or if the image under analysis is big, this parameter can change, or can even be designed as non-uniform, giving more angles at high frequencies and less angles at low frequencies. The scaling parameters are uniformly distributed on a logarithmic scale from one up to an eighth of the size of the image, with a resolution of one third of octave. The maximum scale has been chosen so that at least 90% of the atom energy lies within the signal space when it is centered in the image. Experimentally, it has been found that this scale and rotation discretization choice represents a good compromise between the size of the dictionary, and the efficiency of the representation. One can choose a finer resolution for scale and rotation, getting generally more accuracy in the initial approximations. There is however a price to pay in terms of atom coding and search complexity. Finally, to further constrain the dictionary size, the atoms are always smaller along the second derivative of the Gaussian function than along the Gaussian itself, thus maximizing the similarity of the dictionary elements with edges in images. An example of a dictionary atom can be seen in Figure 6.1.

6.2.4 Benefits of anisotropy

The benefits of anisotropy have been theoretically introduced in Chapter 3. Anisotropy and rotation represent the core of the design of this dictionary, and hence clearly differentiate it from the previously proposed image approximation schemes. To show the gain of using anisotropic refinement, the Anisotropic Refinement atoms dictionary has been compared to four different dictionaries, in terms of the quality of the MP expansion. The first dictionary uses the real part of oriented Gabor atoms generated by translation (\vec{b}), rotation (θ) and isotropic scaling (a) of a modulated Gaussian function, similarly to the one used in [7]:

$$U(a, \theta, \vec{b})g(\vec{x}) = \frac{1}{a} g(a^{-1}r_{-\theta}(\vec{x} - \vec{b})), \quad (6.10)$$

with

$$g(\vec{x}) = e^{i\vec{\omega}_0 \cdot \vec{x}} e^{-\|\vec{x}\|^2/2}. \quad (6.11)$$

The next dictionary is an affine Weyl-Heisenberg dictionary built by translation, dilation and modulation of the Gabor generating atom of Eq. (6.11), as presented in [26]:

$$U(a, \vec{\omega}, \vec{b})g(\vec{x}) = \frac{1}{a} e^{i\vec{\omega} \cdot (\vec{x} - \vec{b})} g(a^{-1}(\vec{x} - \vec{b})), \quad (6.12)$$

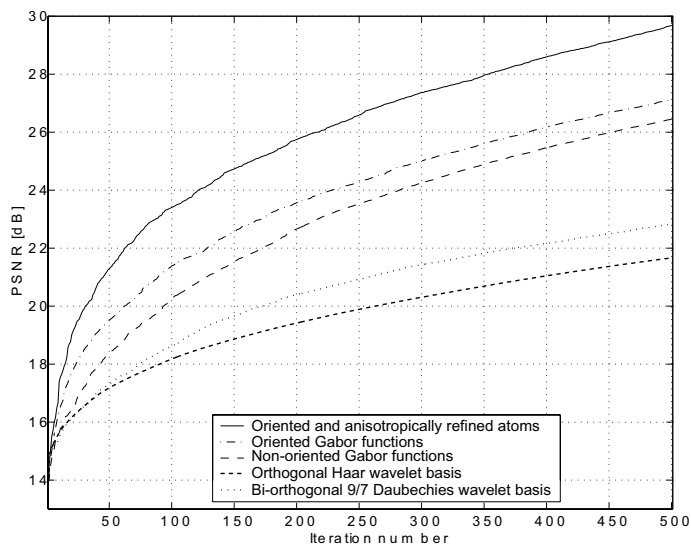


Figure 6.2: Comparison of the approximation properties of different dictionaries of functions: orthogonal Haar wavelet basis, bi-orthogonal Daubechies wavelet basis, non-oriented Gabor functions, oriented Gabor functions and oriented and anisotropically refined atoms.

where, again, only the real part is used. The other two dictionaries are simply built on orthogonal wavelet bases. Figure 6.2 shows the reconstructed PSNR as a function of the number of iterations in the MP expansion using different types of dictionaries. In this figure, the comparison is performed with respect to the number of terms in the expansion, in order to emphasize the approximation properties. Clearly, over-completeness and anisotropic refinement allow to outperform the other dictionaries, in terms of approximation rate, which corresponds to the results presented in [42, 64]. As expected, the orthogonal bases offer the lowest approximation rates due to the fact that these kinds of bases cannot deal with the smoothness of edges. We can thus deduce that redundancy in a carefully designed dictionary provides sparser signal representations. This comparison shows, as well, that the use of rotation is also of interest since the oriented Gabor dictionary gives better results than the modulated one. It is worth noticing that rotation and anisotropic scaling are true 2-D transformations: the use of non-separable dictionaries is clearly beneficial to efficiently approximate 2-D objects. Separable transforms, although they may enable faster implementations, are unable to cope with the geometry of edges.

6.2.5 Representation of the low frequencies of the image

The atoms described in this section are zero-mean, and thus, cannot represent the low frequencies of the image. In a first step, this low frequencies are simply subtracted to the image, and represented independently, just as a downsampled version of the original image, after filtering it with a Gaussian-like filter (the Gaussian filter is approximated by a third order spline).

6.3 GA based MP: a Weak MP implementation

The straight-forward computation of all scalar products required by FS-MP is a computationally hard problem. In order to reduce the computation time, some clever optimization strategy may be taken. Due to the non-linearity and non-convexity of the problem, standard minimization algo-

gorithms, such as steepest descent, do not perform well: they are most likely to fall in one of the many local minima of the problem. One way of implementing a quicker computation of a global solution is through the use of Genetic Algorithms (GA) [31]. The use of a Genetic Algorithm to compute

$$\sup_{\gamma_n \in \Gamma} |\langle \mathbf{R}^n \mathbf{s}, \mathbf{g}_{\gamma_n} \rangle|, \quad (6.13)$$

where Γ is the set of functions in the dictionary, causes the algorithm to find, instead of the optimal solution, a suboptimal one:

$$|\langle \mathbf{R}^n \mathbf{s}, \mathbf{g}_{\gamma_n} \rangle| \geq \alpha \sup_{\gamma_{\text{opt}_n} \in \Gamma} |\langle \mathbf{R}^n \mathbf{s}, \mathbf{g}_{\gamma_{\text{opt}_n}} \rangle| \quad \text{with } 0 < \alpha \leq 1, \quad (6.14)$$

where $\mathbf{g}_{\gamma_{\text{opt}_n}}$ represents the optimal one term approximation of $\mathbf{R}^n \mathbf{s}$, and \mathbf{g}_{γ_n} is the chosen instead. α represents the suboptimality of the solution (how far is the found solution from the optimal one). The implementation of the Greedy algorithm with this suboptimality criteria is called a Weak Greedy Algorithm or Weak MP. Weak Greedy algorithms (see [182] and Chapter 4) have been widely studied [182, 184], and they are proved to converge, even though with slower convergence rate than MP or pure Greedy algorithms.

The implementation of Weak MP through a Genetic Algorithm seems appropriate, because Genetic Algorithms can converge no matter the number of local minima in the problem. The disadvantage of this kind of algorithms is that the convergence degree is statistical. One can never know how close to the optimal solution is the one retrieved. However, greedy algorithms have been demonstrated to converge, even if the solution chosen at every iteration is not the optimal [182, 184], with a statistical optimality degree. Hence, the use of a Genetic Algorithm will not cause the algorithm to diverge, but it will just slow down the convergence.

The use of Genetic Algorithm (GA) in the framework of MP for image approximations was proposed in [61, 65, 66, 72]. The GA takes the parameters of the atoms (translation $\mathbf{b} = [b_x, b_y]$, scaling $\mathbf{a} = [a_x, a_y]$ and rotation θ) as *genes*. Each group of five parameters γ (or five genes $\gamma = [b_x, b_y, a_x, a_y, \theta]$) that define an atom is an *individual*. The group of N_{ind} atoms being evaluated in a certain moment, or the group of N_{ind} individuals alive in a certain moment $[\gamma_1, \gamma_2, \dots, \gamma_{N_{ind}}]$ is a *population*. The individuals of a population will have descendants through *crossover* (mixing of two genes), *mutations* (random variation of the genes) and *survival of the fittest* (keeping the fittest individual of a given generation unchanged in the following generation), which will form the new *generation* of the population. Normally the algorithm will go on until a certain number N_g of generations is achieved or until the minimal error has been reached. In addition, to avoid local minima, a flush of the individuals in a generation (elimination of the individuals) and a regeneration of the population with new genes is performed if a too stable population is achieved. In such a case, only the fittest is kept, and the rest of the individuals of the next generation are randomly initialized*. A diagram of how this algorithm works is depicted in Figure 6.3. The reader interested in a more detailed description of this Genetic Algorithm can check [61, 65].

The computational complexity of the GA directly depends on the number of individuals in the population and on the number of generations allowed. The number of scalar products to compute in order to obtain an MP coefficient through the GA is at most $N_{ind} \cdot N_g - N_g + 1$. From one generation to the other, some individuals can be kept without change due to the randomness of mutations and crossovers. In effect, the computation of the winner is done only once, because then it passes from one generation to the other. If no constrain is given to the dictionary (so, dictionary atoms are not

*A sort of Deluge, where no Noah arch was built, and just one strong individual survived. As it was too sad, being alone, God created some company for him...

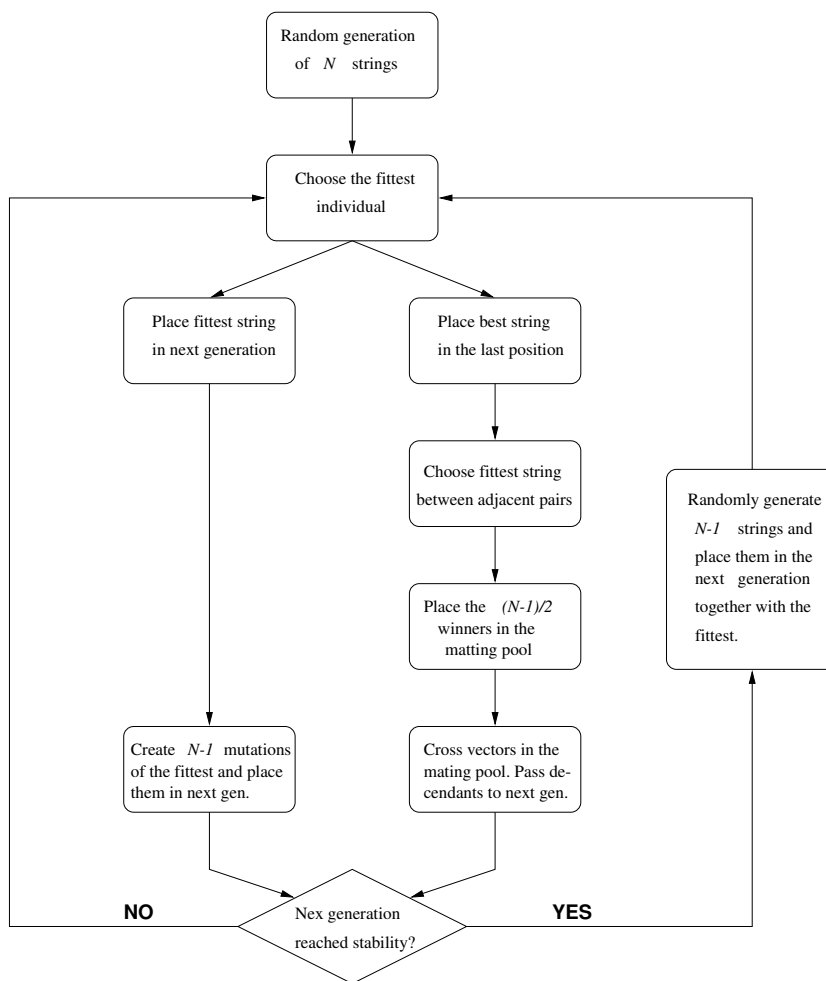


Figure 6.3: Genetic Algorithm block diagram

necessarily bounded, and its size is taken equal to the image size), performing a scalar product will imply N products and sums. The final computational complexity of the GA is given by:

$$O(N \cdot N_{ind} \cdot N_g), \quad (6.15)$$

where N is the dimension of the signal to represent, N_{ind} is the number of individuals in a generation and N_g is the number of generations allowed. When compared to the “Brute Force” Full Search MP (FS-MP) implementation (see Section 6.4.1), this complexity is lower if a small enough number of individuals and generations is taken. Nevertheless, GA introduces a loss of approximation fidelity which is difficult to quantize due to the randomness of the method, and the gain in terms of computational complexity does not always compensate the loss of quality.

6.4 Full Search MP

Sub-optimal algorithms to find the MP coefficient at iteration m can advantageously reduce computational complexity, but there is a price to be paid in terms of accuracy of the solution. This can affect negatively the performance of the application that uses the generated signal expansion. In

fact, errors committed during function selection require further projections in order to be corrected, turning into a decrease of the coefficients decay and, consequently, a reduction of the approximation Sparsity. This can be avoided with the Full Search MP (FS-MP).

For applications where sparsity and accuracy play a critical role, there is a determinate interest in the availability of feasible FS-MP approaches. In this way, the best possible MP expansion may be achieved avoiding heuristic or uncontrolled imprecisions during the function selection procedure. This section presents the “Brute Force” MP algorithm, and states that it is too computationally hard to be used. Then, it introduces some properties of the dictionary that allow to implement a smart version of the FS-MP Algorithm. It compares the approximation results of the FS-MP with the GA-MP, together with the computational complexity and the memory usage of the different MP implementations.

6.4.1 “Brute Force” Full Search MP

According to the MP algorithm principle (see Chapter 4 Section 4.3.2), every iteration needs to compute

$$\sup_{\gamma_m \in \Gamma} |\langle \mathbf{R}^m \mathbf{s}, \mathbf{g}_{\gamma_m} \rangle|, \quad (6.16)$$

where \mathbf{g}_{γ} is the dictionary function and $\mathbf{R}^m \mathbf{s}$ is the residual at iteration m . Thus, it is necessary to browse through all the functions that conform the given dictionary and perform all scalar products between dictionary atoms and the signal. Hence, the complexity of a FS-MP iteration directly depends on the size of the dictionary times the complexity of a single scalar product. In a general case, and without any constraint on the dictionary, the scalar product can be considered to involve all the pixels of the image (non-separable atoms with no compact support). Considering the contribution of the number of functions in the dictionary and the image size, a FS-MP algorithm performs

$$O(N_F \cdot N) \quad (6.17)$$

operations in order to compute all the scalar products, where $N = Size_x \cdot Size_y$ is the dimension of the image and N_F the number of dictionary functions.

The choice of atoms in MP expansions looks forward to find the best spatio-frequential approximation at the best localization within the image to be represented. This implies that the atoms have to be placed at every pixel of the image, as discussed in Section 6.2, in order to take into account all the possible displacements of a certain basis function. This very dense atom translation causes that most of the basis functions in the dictionary are just displaced version of a model, with which they share the frequential properties. The number of basis functions in a dictionary can thus be written as:

$$N_F = M_F \cdot Size_x \cdot Size_y = M_F \cdot N, \quad (6.18)$$

where M_F is the number of model functions. Using the above equation in Eq. (6.17), the complexity of a FS-MP with a dictionary that allows all the integer translations (such as the one used in the scope of this thesis becomes) is:

$$O(M_F \cdot N^2), \quad (6.19)$$

which gives a complexity that grows quadratically with the signal dimension. The above equation does not take into account, in the estimation of the FS-MP complexity, the cost of generating the basis functions that compose the dictionary, since they are just generated once and kept in memory. Nevertheless, in some cases, when the dictionary is generated from an analytic function, such as Eq. (6.2), the programming of the algorithm may be done computing at every step the concerned atom. Then, the complexity of the function generation has to be taken into account to have a

proper estimation of the final complexity of the algorithm. The cost of computing a general atom of the image dimensions is *a priori* related with the size of the image. Assuming Δ the number of operations needed to compute an atom pixel, the complexity of the MP algorithm for one iteration becomes $O(M_F \cdot N^2) + O(\Delta N)$, which, if $\Delta \ll N$, becomes $O(M_F \cdot N^2)$. Nevertheless, in some cases, the supposition that $\Delta \ll N$ cannot be done. This is the case of Eq. (6.2), where there is an exponential to compute. Anyway, the complexity of the algorithm, even when $\Delta \rightarrow N$, remains $O(M_F \cdot N^2)$. If the dictionary atoms have to be computed at every iteration, some tricks can be done to reduce the added complexity this computation implies: The most computationally heavy parts of the analytical atom expression can be kept in look-up tables, which, in case they are defined in a sufficiently dense set of pre-computed values, give a good enough approximation. For example, the $\exp()$ in Eq. (6.2) could be kept in a look-up table, and in case a value that is not in this table is needed, it can be computed through linear interpolation, taking the nearest neighbors that exist in the table.

A solution to save the computational complexity of computing the basis functions at every iteration is to store all the functions that form the dictionary in memory. In such a case, no extra calculations are needed for the MP search, apart from the scalar products and the memory transfers. This is equivalent to disposing of a numerically defined dictionary. It has to be taken into account, though, that the complexity problem of computing all the atoms is completely turned into a memory problem in terms of size and bandwidth. In a general framework with no constraints on the dictionary, saving all the dictionary functions in memory would require $O(N_F \cdot N)$ memory words, which can absolutely be impractical for a very dense dictionary.

Coming back to the dictionary presented in Section 6.2, the fact of having $O(N)$ functions per each of the M_F frequency models represents $O(M_F \cdot N^2)$ in terms of memory requirements. Thus, it becomes absolutely impracticable to keep all the dictionary in memory. On the other hand, the dictionary functions that are not truncated are translation invariant and it is only necessary to store a single numerical centered version for each one of the M_F models. A simple convolution solves then all the scalar products for each one of the frequency models at all the possible integer value translations. Next section gives the detail of this convolution, explaining the special case of boundary functions. Functions near the border are truncated, and this case has to be treated separately.

6.4.2 Spatial invariance in scalar product computation and boundary renormalization

As already said in previous section, at iteration n , the FS-MP has to perform the following operation:

$$\langle \mathbf{R}^m \mathbf{s}, \mathbf{g}_{\gamma_m} \rangle = \sup_{\gamma \in \Gamma} \langle \mathbf{R}^m \mathbf{s}, \mathbf{g}_{\gamma} \rangle. \quad (6.20)$$

Considering the translation invariance of each frequency model, one can write $\langle \mathbf{R}^m \mathbf{s}, \mathbf{g}_{\gamma} \rangle$ as:

$$\langle \mathbf{R}^m \mathbf{s}, \mathbf{g}_{\gamma} \rangle = \langle \mathbf{R}^m \mathbf{s}, \mathbf{g}_{\gamma_{M_F}^{dx, dy}} \rangle = \sum_{\tau} \sum_{\lambda} \mathbf{R}^m \mathbf{s}(\tau, \lambda) \mathbf{g}_{\gamma_{M_F}}(\tau - dx, \lambda - dy), \quad (6.21)$$

where $\gamma_{M_F} \in \Gamma_{M_F}$ and Γ_{M_F} is the sub-set of different frequency models in Γ . Γ_{M_F} is the set of all the possible different shapes that the functions of the dictionary may have independently of the spatial position. In (6.21), $\gamma_{M_F}^{dx, dy}$ indexes all the different functions included in Γ : every possible translation (dx and dy) for all the different frequency models contained in Γ_{M_F} . The, if the signal has an infinite support, by multiplying the Fourier transform of the functions in Γ_{M_F} by the Fourier transform of the image to represent and performing the inverse Fourier Transform, one get the scalar product of the functions analyzed for all the possible translation dx and dy . The fact of being

able to compute the scalar product of a signal through the convolution is only true if the signal domain is infinite. A common problem to be addressed in image processing is the finite nature of the signal. Images are defined on a finite domain. If this is not taken into account, then sparsity in representations may decrease. An example is the representation of an image through the classical wavelet transform. When no considerations are taken on the boundaries (simple zero padding), there will be a large amount of highly valued wavelet coefficients on the image border. Several solutions have been studied to avoid having these non-zero coefficients due to the image boundaries. The most common solutions are periodization of the signal, mirror folding or the use of specific boundary wavelets [118]. Nevertheless, this kind of solutions are not necessarily the best ones, when dealing with redundant dictionaries, and the problem requires a more detailed analysis.

An image $\mathbf{I}(x, y)$ defined in a domain $[x \in [0, Size_x], y \in [0, Size_y]]$ can be considered as the windowed version of a infinite 2-D signal \mathbf{I}_{inf} :

$$\mathbf{I}(x, y) = \Pi \left(\frac{x - Size_x/2}{Size_x}, \frac{y - Size_y/2}{Size_y} \right) \mathbf{I}_{inf}(x, y), \quad (6.22)$$

where $\Pi(x, y)$ is a squared 2-D window defined as non-zero for $x, y \in [0, 1]$.

When dealing with a redundant dictionary that tries to capture the regularity of contours, the possible use of periodic extensions of the image would definitely have a negative impact in the number of functions needed to represent the boundary: artificial structures would be introduced, with discontinuities that would cause a concentration of non-zero coefficients in the boundaries of the image. Mirroring would be a better choice and the energy spreading on the coefficients would be significantly lessen, but would anyway introduce artificial structures such as bending of oriented features (*e.g.* edges) on the image boundary.

A more elegant solution than trying to transform a finite signal to an infinite one is to consider into the dictionary an additional variant of the functions used to decompose the signal, which take these boundaries into account. The proposed supplementary functions substitute any function that goes outside the boundary of the image, and are nothing else than the windowed version of our anisotropically refined functions. The only constraint that these truncated or windowed basis functions have is that they must still have a unitary norm. This can be done by weighting with a normalization factor the displaced versions that cross the image borders, such that they continue to have norm one. In this way, coefficient expansions continue to have the same properties defined for MP signal decomposition on unitary dictionaries. The contribution of this new class of elementary functions is the ability to catch regular structures of the signal (in the same way as the windowed ones) that suddenly terminate at the image boundary without adding any extra coefficient due to the image border. The scalar product computation can then still be performed through the convolution of the centered and non-truncated basis function with the image to represent, with just a normalization factor in the positions where the basis function being analyzed is truncated. The scalar product generation can thus be reformulated as:

$$\begin{aligned} \langle \mathbf{R}^m \mathbf{I}, \mathbf{g}_\gamma \rangle &= \langle \mathbf{R}^m \mathbf{I}, \mathbf{g}_{\gamma_{MF}^{dx, dy}} \rangle = \\ \sum_\tau \sum_\lambda \mathbf{R}^m \mathbf{I}(\tau, \lambda) &\left\| \frac{g_{\gamma_{MF}}(\tau - dx, \lambda - dy)}{\mathbf{g}_{\gamma_{MF}}(\tau - dx, \lambda - dy) \cdot \Pi \left(\frac{\tau - dx - Size_x/2}{Size_x}, \frac{\lambda - dy - Size_y/2}{Size_y} \right)} \right\|_2. \quad (6.23) \\ \Pi \left(\frac{\tau - dx - Size_x/2}{Size_x}, \frac{\lambda - dy - Size_y/2}{Size_y} \right) &= \sum_\tau \sum_\lambda \mathbf{R}^m \mathbf{s}(\tau, \lambda) \cdot \tilde{\mathbf{g}}_{\gamma_{MF}}(\tau - dx, \lambda - dy), \end{aligned}$$

where $\tilde{\mathbf{g}}_\gamma$ represents the new dictionary of functions, which still is space invariant except for a

weighting factor that depends on the spatial location.

The fact that the scalar product is scale invariant apart from a normalization factor implies that its calculation can be divided into two steps. First, the common part of all functions of the same kind (same non bounded spatio-frequential properties but not same position) is used in the convolution. Afterward, each obtained coefficient is weighted by the normalization factor of its projecting function, which takes into account its intersection with the boundary. The computation of the normalization factors implies an additional increase of complexity. In a general dictionary, this complexity depends on the total number of functions of the dictionary and the size of the signal. Considering the general situation where all dictionary functions may intersect with the boundary independently of its position, it turns out that N_F normalization factors are needed and

$$O(M_F \cdot N^2) \quad (6.24)$$

operations can be needed to compute them. This normalizing factors are used at every iteration of the MP algorithm, and do not depend on the signal under analysis. This allows to compute them once at the beginning of the expansion process and to store them for their posterior use in the subsequent MP iterations. When this one time computation and storage is performed, Eq. (6.18) trivially implies the need of

$$O(M_F \cdot N) \quad (6.25)$$

memory words, which in some cases and depending on the dictionary size (and image size) may be feasible.

6.4.3 FFT based FS-MP: from Scalar Products to Spatial Convolution

In Section 6.4.1 the computation of the whole set of scalar products has been presented as a very computationally complex task, while Section 6.4.2 has explained the special case of truncated basis functions, and how these differ from the non truncated ones. In order to lessen the complexity of computing the convolution, the use of a fast convolution algorithm based on the Fourier transform is to be considered. The use of such a fast algorithm makes a full search strategy for large dictionaries possible. Eq. (6.19), which gives the complexity of computing the scalar products through convolution, is then substituted by

$$O(M_F \cdot N \log(N)). \quad (6.26)$$

If the fast convolution using the Fourier Transform is used, it is more efficient to store in memory the Fourier version of the basis functions, such that only one FFT will be needed (and IFFT, to obtain the spatial convolution). In the case of a dictionary of non-complex functions, the amount of memory needed for the Fourier domain versions is the same as the amount needed in the spatial domain. However, as performing a Fourier Transform causes a periodization in space, some spatial zero padding has to be added, in order to avoid this periodization effect give artificial non-zero boundary coefficients. This zero padding has to be big enough, so that the basis function does not see the periodization, but small enough, not to give too much extra computations. In the worst case (basis function with support as big as the image size), the zero padding has to be of $\frac{Size_x}{2}$ pixels per side horizontally, and $\frac{Size_y}{2}$ per side vertically. Furthermore, the mask of normalizing weights has to be stored in memory as well, in order not to compute it at every iteration, which increases even more the memory needs. Taking these facts into account, one gets an asymptotic memory requirement of:

$$O(M_F \cdot N^2) \quad (6.27)$$

memory words.

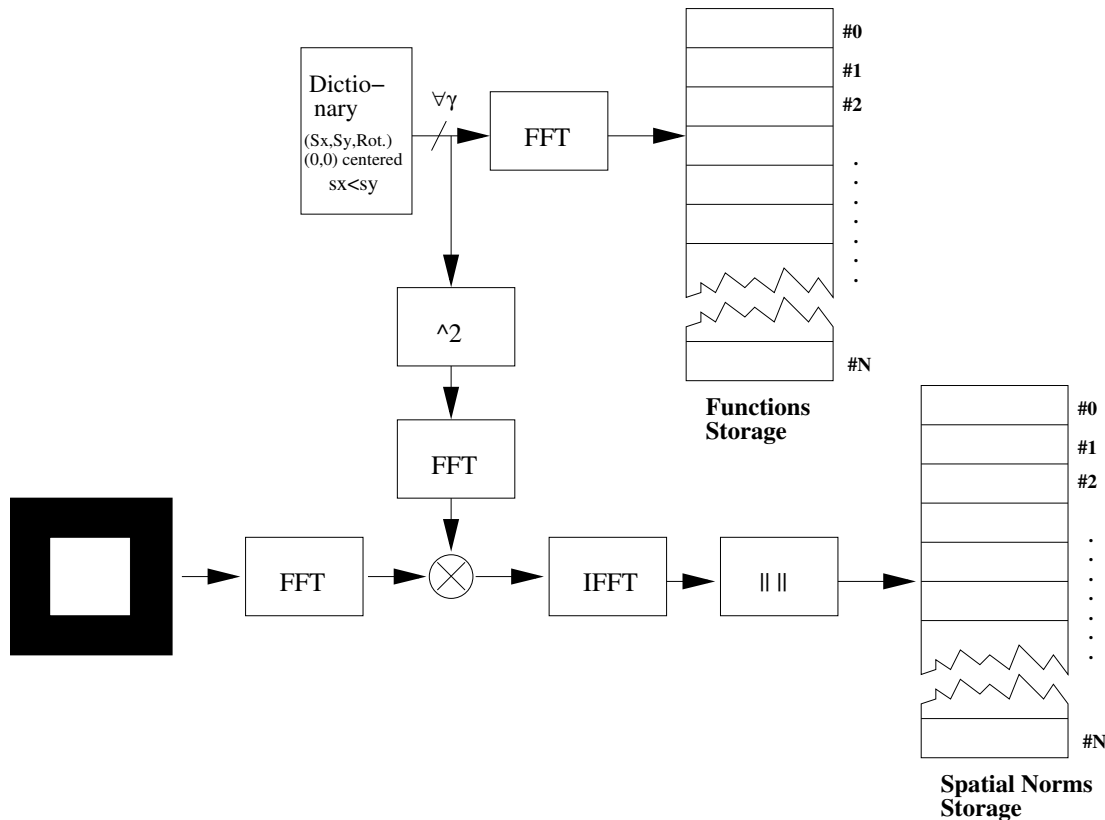


Figure 6.4: Description of the setting up procedure to generate the look up tables to be used in Figure 6.5

The scheme used to generate and store the dictionary functions and the weighting masks is shown in Figure 6.4. The dictionary functions are generated from its analytic expression, and after being transformed in the Fourier domain, stored in a look-up table. Weighting masks are generated by convolving a centered squared atom by an image size pulse function of unitary amplitude. The weighting masks are stored in another look-up table.

The generation of the scalar products from the stored Fourier atoms and their weighting masks is then generated as illustrated in Figure 6.5. Convolutions are performed on the Fourier domain, by multiplying the signal to represent with the basis function. Once this product has been performed, the output is transformed to the spatial domain, where it is multiplied by the weighting mask of the basis function being analyzed. Once scalar products are available, as reviewed in Section 4.3.2, only the search of the suprema of all these scalar products has to be done.

This section has presented a faster FS-MP, using the Fourier domain convolution to compute the scalar products of the signal to represent and the basis functions in the dictionary. It has been seen, though, that even though this scheme is faster than the “Brute force” approach, it still needs high amounts of memory. This approach, though, has not taken benefit from specific dictionary features, such as space-frequency localization. Section 6.5 will present how MP can benefit of very interesting improvements in terms of computational cost and memory requirements when special care of the structure of the dictionary is taken into account. Before trying to optimize the full search algorithm memory and computational requirements, though, it is worth seeing what are the improvements in terms of approximation properties of using a full search algorithm instead of a genetic based MP.

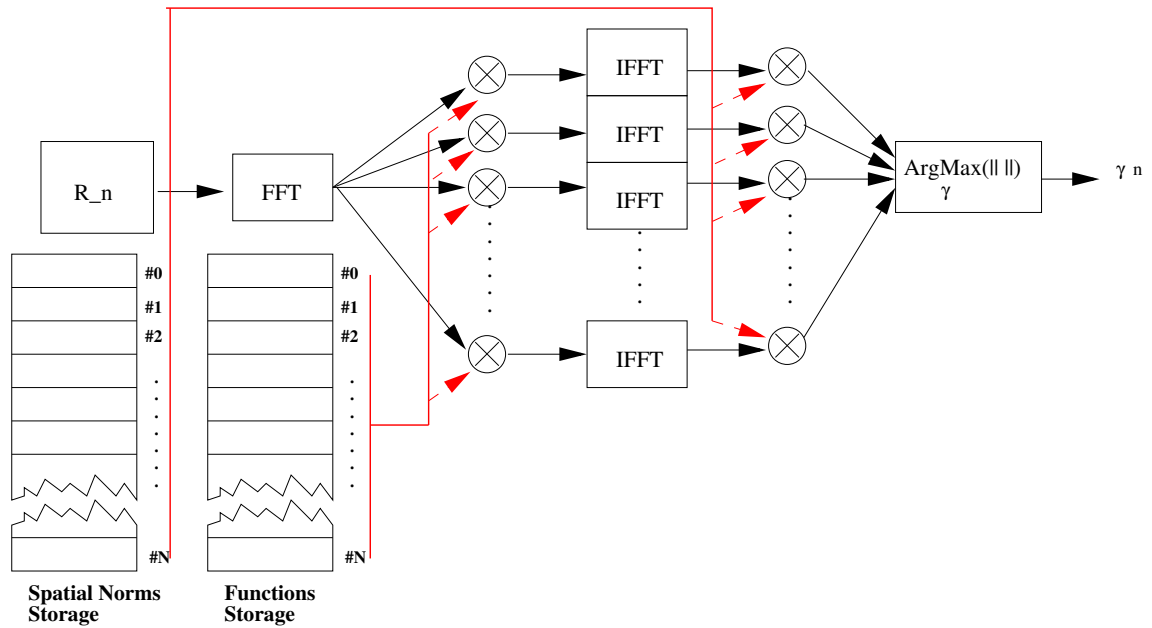


Figure 6.5: Schematic description of the full search algorithm for one iteration of MP. Look up tables are used to hold the DFT of the dictionary functions and the normalizing masks.

Image	Algorithm	Computation Time	PSNR
<i>Lenna</i> 128x128	GA-MP	156' 26.909"	26.1506 dB
	FS-MP	126' 29.792"	29.9076 dB
<i>Cameraman</i> 128x128	GA-MP	157' 3.249"	25.0485 dB
	FS-MP	128' 13.957"	28.8270 dB
<i>Baboon</i> 128x128	GA-MP	160' 19.048"	25.1374 dB
	FS-MP	128' 9.170"	26.7932 dB

Table 6.1: Comparison of the computational time and the image quality obtained with the Genetic Algorithm, using 39 individuals and 75 generations, and the Full Search algorithm.

6.4.4 Results: Full Search vs Genetic Algorithm Search

This section compares the performance of the Full Search MP (FS-MP) algorithm versus the Weak MP based on the GA (GA-MP). Table 6.1 states the relations of computation time required by each algorithm and the approximation quality achieved. The experiments have been performed on a Dual Xeon 2.2 GHz with 2GB of RAM platform. One can see that the optimized full search algorithm improves, for similar or smaller computation times than the GA, the quality of the approximation. It is very relevant the improvement of 3.8 dBs registered in the piecewise-smooth like images (*Lenna* and *Cameraman*) with a significantly smaller computational time. For the case of *Baboon*, the improvement is only of 1.6 dBs due to the higher complexity of the signal (much more textured). In the three cases, approximations have been generated by recovering the first 300 terms of their MP expansions.

Figures 6.7, 6.8 and 6.9 show a visual comparison of the approximation properties of the FS-MP and the GA-MP. The visual comparison of Figures 6.7, 6.8 and 6.9 with their original images (Figure 6.6) shows that the GA causes some *geometric* noise to appear in the image approximations



Figure 6.6: Original *Lenna*, *Cameraman* and *Baboon* images.

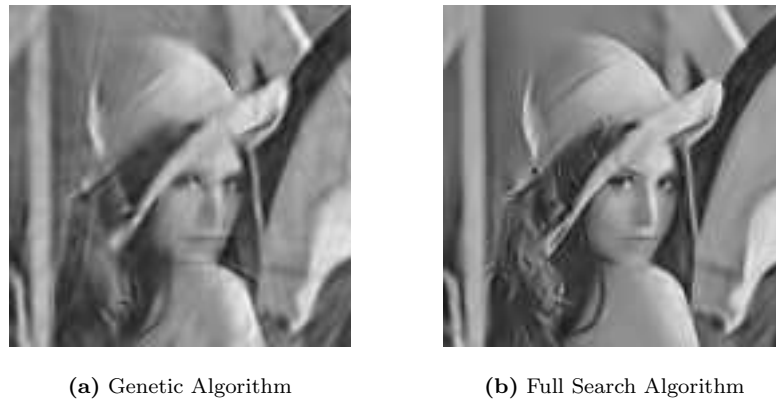


Figure 6.7: Visual comparison of *Lenna* 128x128 decomposed with MP with 300 coefficients (a) with the GA-MP and (b) with the FS-MP.

performed through this implementation of Weak MP. This geometric noise, or lines that extend the contours of an object, appear near relevant salient geometric structures, such as edges, due to the suboptimality of the GA search. In the FS case, the exhaustive search at every iteration drastically reduces this kind of noise.

A numerical comparison about the approximation properties of GA-MP and FS-MP can be found in Figure 6.10. This figure shows the evolution of the PSNR with the number of coefficients for the FS-MP algorithm and the GA based MP algorithm, when the image *Lenna* is approximated. As expected, the FS algorithm outperforms through all iterations the weak greedy algorithm, and this out-performance increases as the number of terms in the expansion increases as well. It is thus clear that the quick implementation of a FS-MP is worth, if one wants to use MP for image approximation, both for the PSNR improvement and the visual quality improvement that it brings.

6.5 Exploiting the dictionary features

In the previous section, the use of the Fourier Transform to perform the convolution and the computational gain that this brings have been studied for a general dictionary. It has been shown

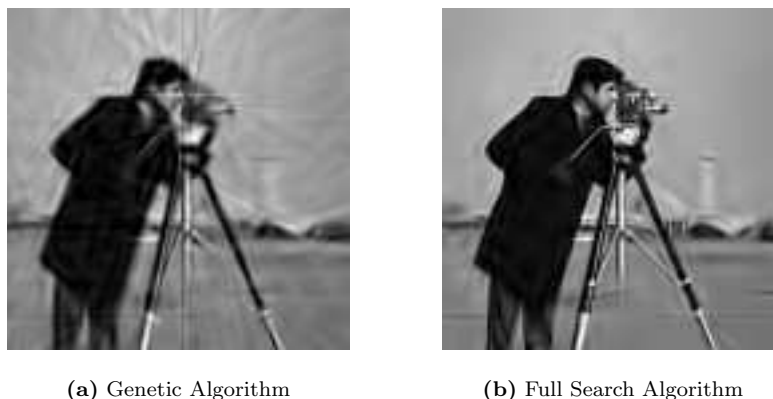


Figure 6.8: Visual comparison of *Cameraman* 128x128 decomposed with MP with 300 coefficients (a) with the Genetic Algorithm and (b) with the Full Search MP.

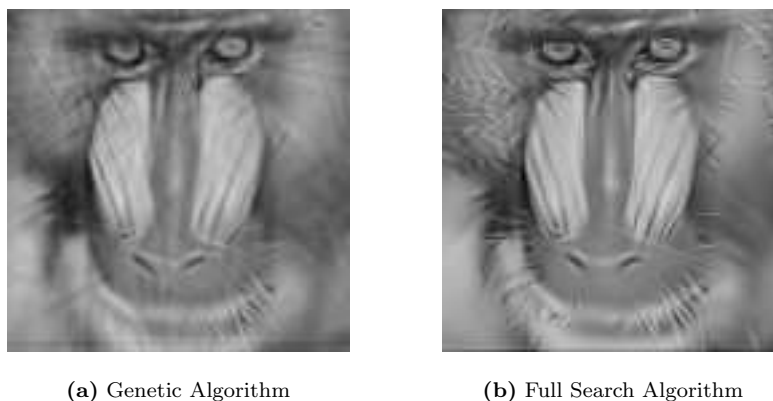


Figure 6.9: Visual comparison of *Baboon* 128x128 decomposed with MP with 300 coefficients (a) with the Genetic Algorithm and (b) with the Full Search MP.

that the fact of using the Fourier Transform to perform a convolution implies reducing the cost of computing a coefficient of the MP expansion by a square factor. However, this reduction could be even higher if some specific features of the dictionary were taken into account. In this section, the dictionary spatio-frequency localization is exploited to reduce both computational complexity and memory usage. Several results are presented to illustrate the gains and the limits of this reduction.

6.5.1 Exploiting Spatio-Temporal Energy Localization: Compact Support and Atoms Approximation

The presented full search approach can easily reduce the computational complexity if the dictionary of basis functions has compact support functions in it. There are basically two different sorts of compactness: compactness in frequency and compactness in space. Natural image decompositions often use dictionaries where functions have spatial localized support, see for example the one proposed in Section 6.2 or, more generally, the basis functions used for classic dyadic wavelet decompositions [118]. Concerning the dictionary used in this work, even if the functions in use do not have

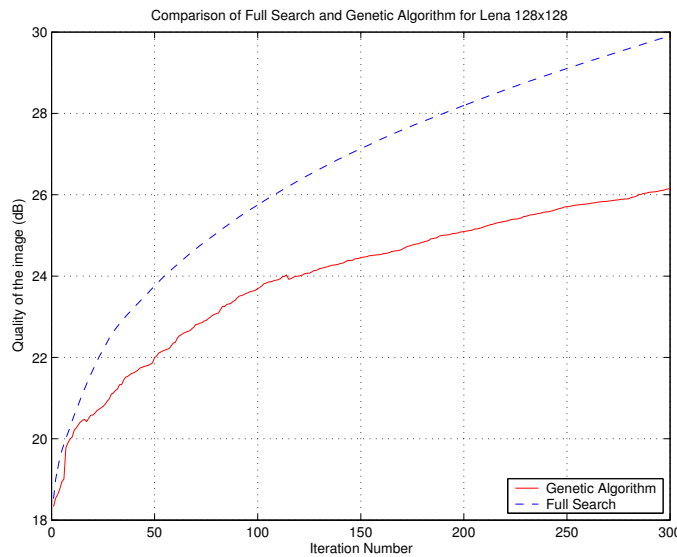


Figure 6.10: Evolution of the PSNR with the iteration number for the Genetic and the FS-MP algorithms. The image used in this case is the gray-scale 128x128 Lena.

a compact support, energy is mostly compacted on a very localized fashion. Moreover, as atoms envelop is Gaussian shaped, energy is localized in both domains: space and frequency. This allows to perform an approximation of the dictionary which significantly reduces the memory usage and the computational resources needed, without affecting the MP rate of convergence. This section presents such a computational optimization, and its limits*.

A first advantage of spatially localized dictionaries is that one can use the so called M -fold MP strategy [79]. This consist in recovering M atoms at each MP iteration. Indeed, distant atoms in space that have a localized support, are often incoherent among them. Some heuristics can be based on this and contribute with M times speeds-up.

Thanks to the incoherence among spatially distant atoms, further savings are possible. Indeed, for a given MP iteration, once an atom has been selected, all the scalar products of those atoms that are highly incoherent with the selected one do not need to be recomputed. This can bring to the possibility of parallelizing the MP, and computing it for different influence zones. This strategy can be in some cases also coupled with the M -fold approach leading to an even better general performance.

Even though these two approaches (M -fold MP and using the spatial incoherence of the dictionary) can improve the computational time needed, they are not discussed in this chapter. The full search solution proposed here is based on the massive use of the FFT in order to reduce the complexity of computing the scalar products of all the atom translations. For this purpose, all the centered versions of the atoms in the Fourier domain are stored into memory such that they are only computed once. The fact that an atom and the normalizing modulus masks (as described in Section 6.4) may have few non-zero samples in an image gives the possibility to optimize their storage by efficiently indicating where zero samples are, reducing the amount of memory needed for the MP full search algorithm. Moreover, the efficient description of zero sample positions can also be used to reduce the number of multiplications involved in the algorithm.

*Original idea and implementation of Oscar Divorra Escoda.

Memory Compression

The reduction of the memory usage of the MP algorithm is performed by taking profit of the localization in frequency of the atoms used. Figure 6.11 illustrates the kind of information contained in the maps that represent the frequency domain form of atoms (see the first row), and the normalizing factor maps (see the third row), used to compensate the image border effects in the scalar products computation. The second and fourth row of Figure 6.11 represent, respectively, for the Fourier domain atoms and normalization masks, the areas of each map where relevant information is cumulated (white areas). Concerning the Fourier domain atoms version, in this examples one sees in the second column, in white, the support where the modulus of the Fourier transform is greater than a given threshold (10^{-3} in this example). In the fourth column, the areas in white are the locations where the renormalization factor differs from 1 by more than a certain threshold -10^{-3} in the presented case – (factors smaller than 1 are used to re-normalize scalar products of atoms that significantly cross the image border). In both cases, only the original values that correspond to the white areas are required to be stored, the rest can be approximated by zero for the first case or one in the second case without a significant lost of quality.

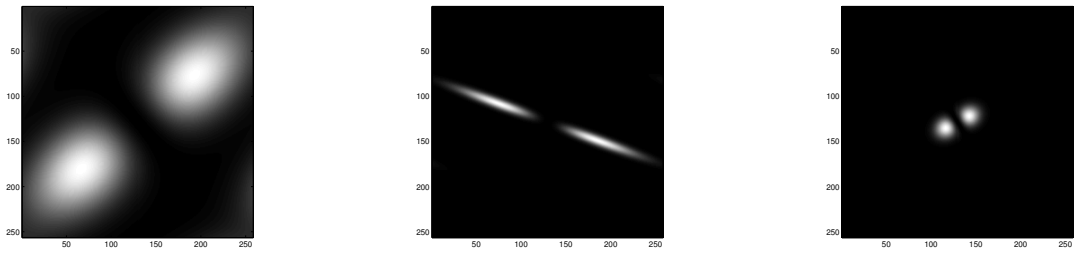
It is interesting to notice, in Figure 6.11, that zeros and ones cumulate in successive memory position. Such a case distribution of significant and non-significant values allows an easy reduction of the storage space needed, by using a basic run-length coding [164] to store more efficiently into memory the default values (zero or one) for every kind of map. Of course, better techniques using quadrees could be used, but they would slow down the algorithm, and thus they are out of the scope of this work.

Figure 6.12 depicts the scanning order followed in the run-length based storage of a simple example. In a top-down fashion, the signal is scanned line by line. Two lists of values are kept: one contains the list of elements belonging to every run of significant pixels and the other contains how long is each run of values (significant and 1 or 0 constant).

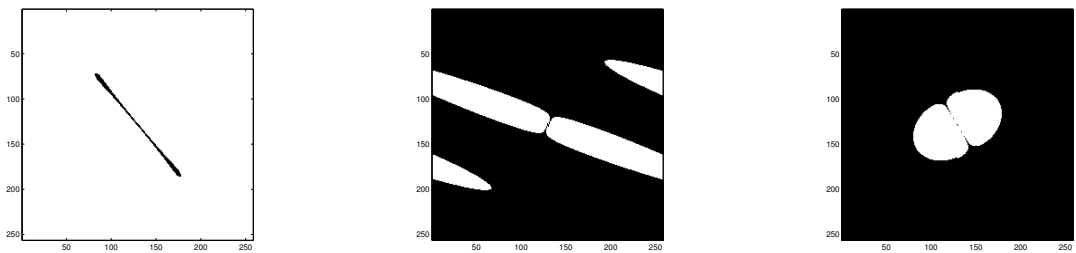
Examples

The fact of approximating to zero or to one some values that are close to these values has, of course, some impact in the quality of the reconstruction and on the atom shape. This impact has to be analyzed, in order to check that it does not negatively affect the quality of the decomposition. Figures 6.13 and 6.14 show how memory compression affects the quality of the approximation of Lenna (gray-scale 128x128 pixels) when using FS-MP. Figure 6.13 plots the memory occupation versus the quality obtained for a *Lenna* approximation of 100 terms and a give threshold of approximation in the memory compression scheme. Figure 6.14 shows some visual results with the same method. The degree of memory compression is varied by means of changing the threshold used to decide which region has significant values. The experiments are done by logarithmically varying the threshold from zero up to 100 ($0, 10^{-5}, 10^{-4}, 10^{-3}, \dots, 100$). This means that, for the Fourier modulus of the basis functions, everything that is smaller than the threshold is set to zero, and for the masks, every $1 - \text{mask_value}$ that is smaller than the threshold is set to zero as well. Notice that, for the mask, having a threshold bigger than one simply means ignoring the renormalization of the truncated basis functions.

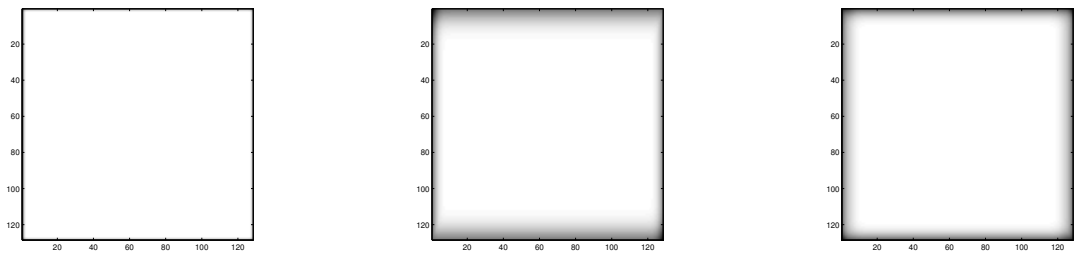
As can be appreciated in the figures, the dictionary can be approximated such that up to more than 75% of memory is saved without any significant increase in distortion. Figure 6.14 shows that visually, the threshold effects appear even later than when looking at the PSNR plot. A very high threshold has to be set before important distortions are perceived.



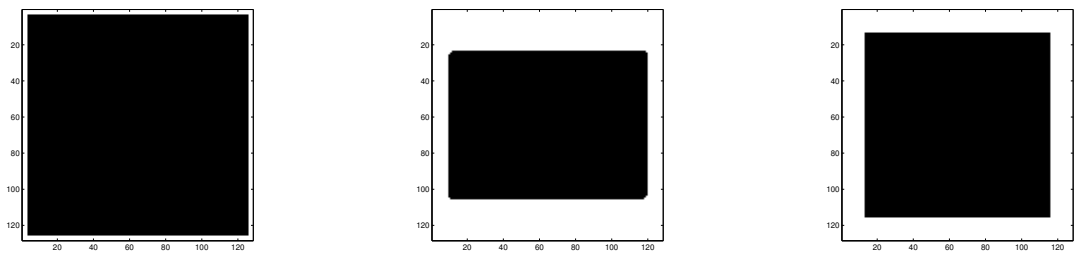
(a) Atoms frequency modulus



(b) Atoms support



(c) Renormalization maps



(d) Binary renormalization mask

Figure 6.11: Example of different functions in the dictionary and their respective normalization masks. (a) Frequency modulus of three selected dictionary atoms (white corresponds to high values and black to zero). (b) Respective support of the significant values that need to be stored in memory (in white, values with modulus greater than 0.001). (c) Spatial renormalization maps that correspond to the atoms of the first row (white corresponds to 1 and black to zero). (d) White zones represent the binary mask that determines where values different from 1.0 with a significant difference (greater than 0.001) are located.

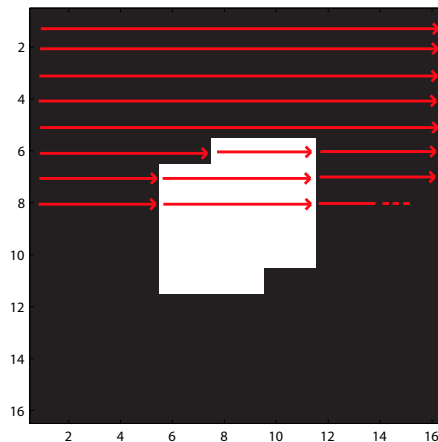


Figure 6.12: Every Fourier domain atom is stored in a run-length fashion. All values considered non-significant (black zones in the figure) are set to zero. All consecutive zeros (considering a line wise scanning of images, as indicated by the red arrows) are efficiently stored using a single integer. Significant values (white zones in the image) are stored one by one together with an integer number that specifies how many significant values are consecutively aligned.

6.5.2 Steerability of Atoms and Complexity Benefits

For the purposes of this work, the FS-MP algorithm with memory compression is already efficient enough, in terms of computation speed and memory occupation. Nevertheless, for some applications, one may be interested in having a quicker and less memory consuming scheme. This can be achieved using steerability.

Steerability is an interesting tool when dealing with orientable functions. It is based on the principle that certain classes of kernels can be tuned according to a set of geometric parameters at the same time that can be decomposed in a linear combination of few functions sampled from that parameter space. For example, for the kind of atoms used in this work, when they have isotropic scaling, it is possible to generate all the possible orientations of these from just 3 basic functions with different orientations [69]. It is common to use the orientations of 0 , $\frac{\pi}{3}$ and $\frac{2\pi}{3}$ (see Figure 6.15). However, the concept of steerability can be extended to any other affine transformation. In some cases, the steered functions are only approximations of the ideal ones. An extensive review of steerability can be found in [69, 94, 95, 120, 145].

In terms of computational advantages, the use of steerability would contribute significantly to reduce the number of necessary inverse Fourier transforms. To give an example, let us take again the isotropically scaled second derivative of a Gaussian. Consider that the scalar products of all the translated versions of N_θ different orientations of our steerable kernel are needed. For that purpose, if we store into memory a Fourier domain version of the function for each one of the N_θ orientations, as described in Section 6.4, a total number of:

$$O(N) \cdot N_\theta + O(N \log N) \cdot N_\theta \quad (6.28)$$

operations will be necessary (where N is the size of the data). In Eq. (6.28), the first term corresponds to the product in the transformed domain for filtering, and the second indicates the complexity associated to all the inverse Fourier transforms. If steerability is used instead, and all the needed

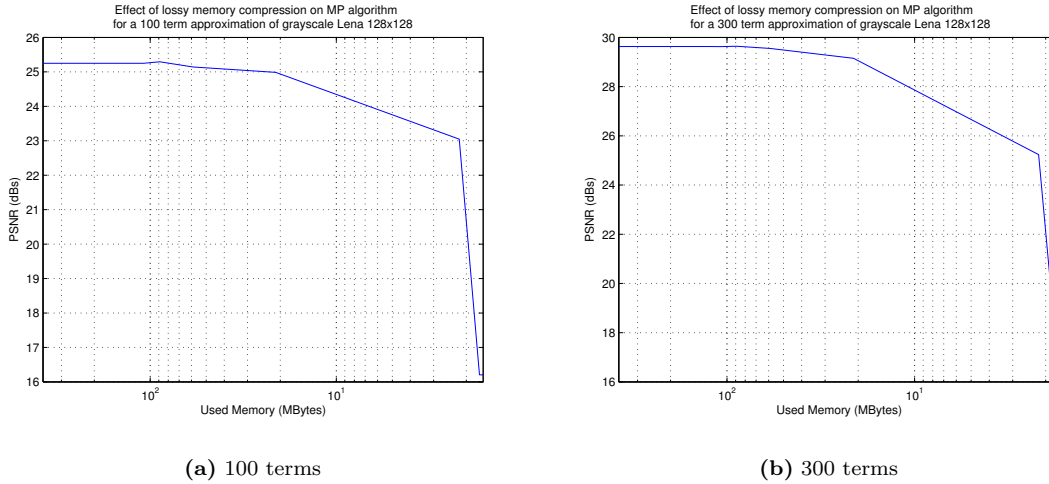


Figure 6.13: Approximation PSNR vs memory used for the FFT based full search algorithm for Lena 128x128. In the left the approximation is done with 100 terms. In the right, 300 terms are used. The graphics have been obtained by increasing the threshold of approximation logarithmically. It can be seen that up to 75% in memory savings can be achieved without loss in approximation convergence.

orientations are obtained by linear combination of a basic set composed by three functions, the number of necessary operations turns into:

$$O(N) \cdot 3 + O(N \log N) \cdot 3 + O(N) \cdot 5 \cdot (N_\theta - 3), \quad (6.29)$$

where the first two terms correspond to the complexity required by the Fourier domain filtering step and the last term corresponds to the steerability based computation of the remaining orientations. The 5 factor is due to the three products and two additions of the linear combination used to steer the filters. Figure 6.16 illustrates the potential savings of using steerability in the computation of the scalar products for the simple example of isotropically scaled anisotropic second derivatives of a Gaussian. Even for small N , the use of steerability is computationally advantageous. Moreover, the bigger the N the more resources are spared. According to the particular example given by (6.28) and (6.29) for $N_\theta = 36$, when $N \rightarrow \infty$ the spare factor converges to 12. Moreover, in terms of memory usage, the amount of needed storage falls down of a factor $\frac{N_\theta}{3}$. This not only reduces the memory consumption but also the time required to transfer the data from memory to the processor.

This section has stated the advantages of introducing the steerability principle in the analysis step of each MP iteration. Nevertheless, its practical implementation and integration is out of the scope of this work, and will be left for future work.

6.6 MP with low-pass and band-pass dictionary

Until here, an image approximation method that uses a highly redundant dictionary has been presented. The dictionary is based on geometric transformation (translation, anisotropic scaling and rotation) of a single basis function. It has been seen that the approximation properties of such an algorithm are very good, with few terms, one gets approximations with a small residual and good visual quality. But the image approximation scheme presented here is not still fully based on the use



Figure 6.14: Visual comparison of the different approximation of Lenna 128x128 for the amounts of memory of: (a) 377 MB (threshold 0), (b) 122 MB (threshold 10^{-4}), (c) 22 MB (threshold 1) and (d) 2.2 MB (threshold 10).

of redundant dictionaries, the low-frequencies of the image need to be treated separately. One of the main advantages that the use of fully geometrically based image representations is the possibility of performing geometric image manipulations (such as reconstruction at different size or rotation of the image) without the need of interpolation. This flexibility is not present in the approximation scheme that uses a downsampled version of the low frequency image in order to represent this part of the visual information. It is interesting, then, to design a dictionary of basis functions that takes into account the low frequencies of the image as well through geometrical transformations of a single basis function. In order to be able to keep on using the FS-MP algorithm, this basis function should have similar characteristics than the Anisotropic Refinement atoms, defined by Eq. (6.2). This section presents such a dictionary, and explains how it has been designed, in order to be able to capture all the frequencies of the image.

As in the Anisotropic Refinement Atoms, the low pass atoms are chosen to have Gaussian support, to have spatial and frequential localization. Gaussians are already low-band pass, and thus, no more modulation or derivation is needed. The generating function is then:

$$g(x, y) = C e^{-x^2 - y^2}, \quad (6.30)$$

where $C = \sqrt{\frac{2}{\pi}}$ is a normalizing constant. Of course, the low-pass atoms need to be translated all over the image as well, similarly to the Anisotropic refinement atoms. They also need of dilations,

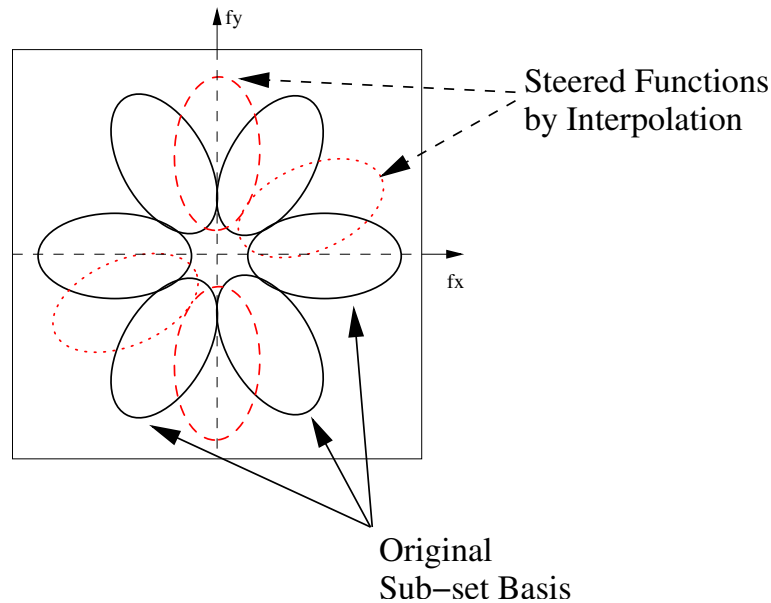


Figure 6.15: In some cases, a whole set of functions from the dictionary can be generated from the linear combination of a subset of these. This is the case, for example, for isotropically scaled Gaussian second derivatives [69], only three real filters (drawn in continuous line) are needed to generate all the rest of orientations.

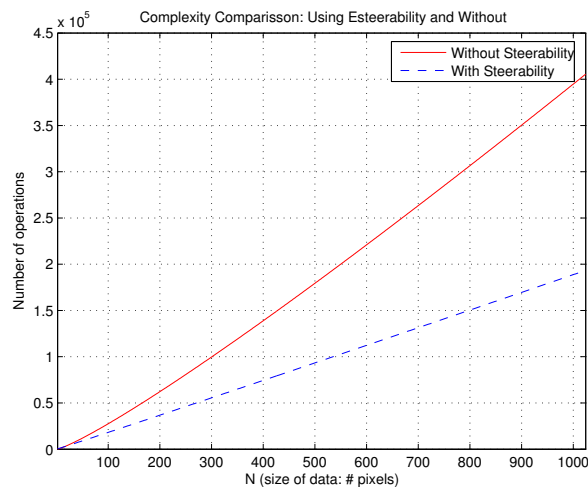


Figure 6.16: Comparison of the growth of complexity with and without using steerability as a function of N . This graph has been performed assuming $N_\theta = 36$.

in order to be capable of capturing big or small smooth zones. As big Anisotropic Refinement atoms already capture middle frequencies, anisotropy and rotations are not included in the low-pass dictionary. Elongated and thin structures are already properly captured by Anisotropic Refinement Atoms. Thus, the dictionary of basis functions have now two generating functions in it, one for the low frequencies of the images (big smooth zones) and one for the middle to high frequencies (basically, contours). The final dictionary of functions that will be used to decompose an image is

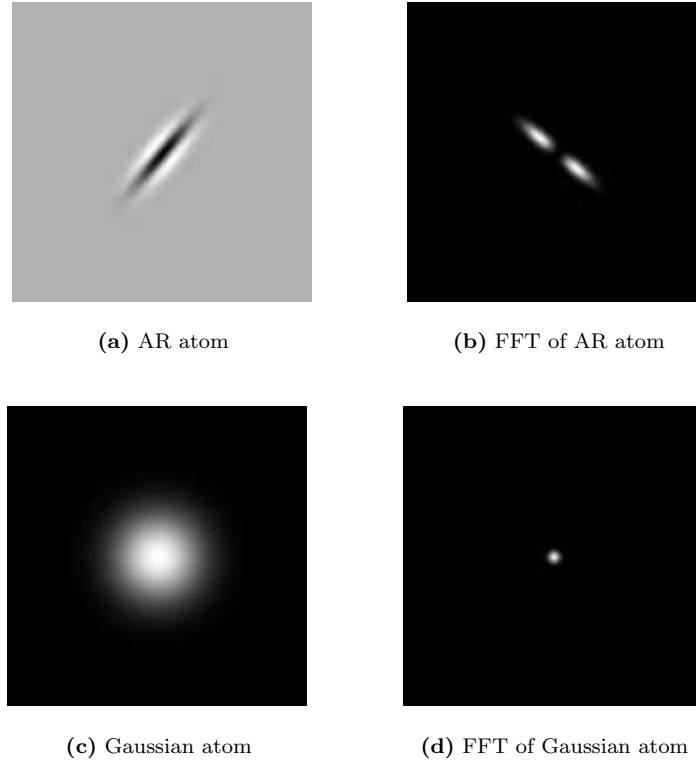


Figure 6.17: Example of AR and Gaussian atoms and their respective Fourier Transforms.

then formed by two subdictionaries: the Anisotropic Refinement Atoms dictionary, generated by translation, anisotropic scaling and rotation of a wavelet like basis functions (see Section 6.2), and the Gaussian Atoms dictionary, generated by translating and isotropically scaling a Gaussian function. An example of atoms belonging to these two families of functions can be seen in Figure 6.17.

6.6.1 Construction of the dictionary

Again, the dictionary of basis functions is constructed by applying a unitary group to, in this case, to the two generating functions defined before:

$$g_1(x, y) = \sqrt{\frac{2}{\pi}} e^{-x^2 - y^2}, \quad (6.31)$$

$$g_2(x, y) = \sqrt{\frac{2}{3\pi}} (4x^2 - 2) e^{-(x^2 + y^2)}, \quad (6.32)$$

Now, the question is how dense these dictionaries have to be, and which sampling of the transformation parameters has to be done in order to ensure a dictionary redundant enough to have efficient approximations, but small enough to be able to compute these approximations with a finite time.

The Anisotropic Refinement Atoms have the same sampling of the parameters that has been exposed in Section 6.2, with an angle resolution of $\frac{\pi}{18}$ and the scales octave-based sampled, with three scales per octave. The translations take any integer value going from zero to the image size (pixel accuracy translation).

The Gaussian dictionary does not have anisotropy or rotations, it has only isotropic scaling and translations. The translation can also take any integer value from zero the size of the image.

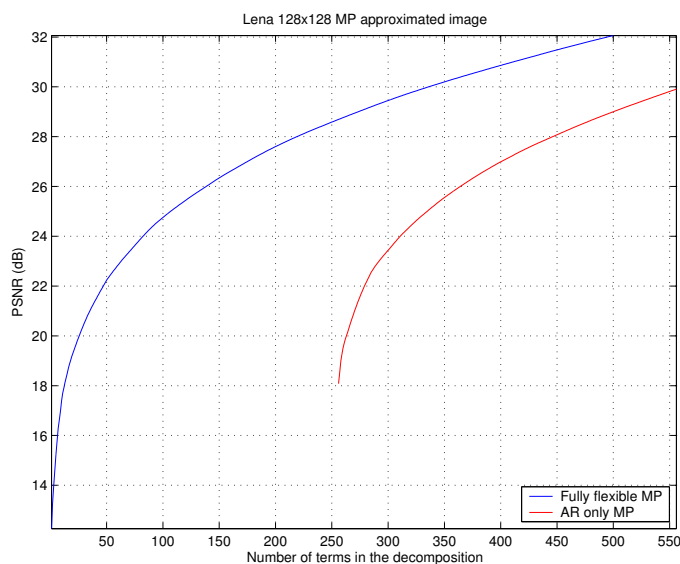
The dilations are computed so that there is some overlap in the frequency domain between the Anisotropic Refinement atoms and the Gaussian dictionary, but that this overlap is small enough so that there are not destructive interactions between the two dictionaries. The sampling of the scaling factors is taken, similarly to the case of the anisotropic refinement atoms, octave based, with at least three scales per octave. The maximum dilation of this dictionary is taken so that at least a 90% of the energy of the atom is inside the signal domain when this is placed in the center. A more detailed computation of the maximum and the minimum scaling factors allowed for every dictionary can be found in Appendix B.

6.7 Fully flexible image representation with MP

When decomposing an image with the dictionary described in the previous section, the image is fully represented in a set of basis functions that are covariant to translations, rotations and dilations. In addition, the approximation has the coefficient upper-bound that is strictly decreasing. This gives a fully flexible image approximation, in the sense that one can truncate this approximation and have an approximation with fewer terms, or, from an approximation of n terms, continue and have an approximation with $n + t$ terms. In addition, thanks to the properties of the set of functions used, which are covariant to translations, dilations and rotations, a reconstruction of a different size image or a rotated images is straightforward. The covariance to dilations is specially useful, because it allows to reconstruct images at a bigger or smaller size than the original image without having to perform any interpolation operations. Some examples of images reconstructed at different sizes are given in Chapter 7. A way to see this property is through the invariance to dilations. The covariance to dilations can be understood in a very intuitive way if one thinks that the dictionary is formed by continuous functions, that have been sampled to a certain rate, in order to use them in discrete signals. Changing the dilation factor of all the functions that form an image decomposition can be seen as changing the sampling rate. If one divides all the scaling factors of an image by two, it corresponds to sampling at a frequency two times smaller, and thus, one gets half of the points it had. On the contrary, if one multiplies all the scaling factors of all the functions that form an image decomposition, it is equivalent at multiplying by two the sampling frequency, and thus, the size of the image will double (one gets a number of points that is the double of the original image).

Both dictionaries, the high-pass and the low-pass are created by performing affine transformations to a basis function. As both the low-pass and the high-pass dictionary functions are continuous functions covariant to dilations, rotations and translations, the dictionary gives a continuous representation, covariant to these affine transformations. In addition, the AR atoms are covariant to rotations, and the low-pass atoms are invariant to them (they do not have rotation parameter). This causes the dictionary to be covariant to rotations as well. The image represented with such a dictionary has a fully continuous representation, and can be thus reconstructed at any scale or rotations only by varying the parameters of the atoms that represent the scale or the rotation. When using the low-pass image version instead of the Gaussian dictionary, this was not possible, due to the fact that the low-pass image did not have a representation that could have an affine transformation applied to it by just changing the rotation or the dilation parameter of the basis function. The low pass image was not continuous, and thus, changing the sampling rate without having to perform an interpolation operation was not possible.

The question is whether the fact of using a Gaussian and an Anisotropic Refinement Atoms dictionaries causes a gain or a loss of approximation properties. The fact of having a fully flexible representation (covariant to translations, dilations and rotations) may have some price to be paid in terms of approximation properties. Nevertheless, this is not the case. One has to count every pixel



(c) PSNR graphic

Figure 6.18: Visual comparison of the image obtained with MP without the low pass with 500 terms (a) and MP with the low-pass dictionary with 500 terms(b), and evolution of the PSNR with the number of coefficients for *Lenna* 128×128 image.

of the low pass image as a basis function, and thus the approximation properties are not lost due to the use of these two subdictionaries, as can be seen in the example given in Figure 6.18, which shows the evolution of the PSNR with the number of terms for *Lenna* 128×128 . Both dictionaries have three scales per octave and 18 angles in the anisotropic refinement part. For the image of *Lenna* obtained with the fully flexible representation, the Gaussian dictionary dilation parameter has been sampled at three scales per function as well. Notice that, to perform this comparison, every pixel in the low resolution image (that for a 128×128 image can be of size 16×16) is counted as a basis function (every pixel is considered to be a delta function). Of course, this graphic shows only the evolution of the PSNR with the number of coefficients. This does not mean that the Rate-Distortion properties of the fully flexible scheme are more performing than for the scheme that does not include the low-band dictionary. The rate distortion properties are discussed in the next chapter.

6.8 MP with *a priori* knowledge: Weighted MP

The image representation scheme presented in this chapter is efficient for edge representation. But natural images have often textured parts, which are not efficiently approximated by the edge like functions. For representations with very few terms, this fact is not so important, because normally textures bring less information, and can be ignored. In some cases, though, one can be interested in coding the textures at the same time as the contours. The efficiency would be improved if a set of basis functions more efficient at representing textures than the Anisotropic Refinement atoms was included in the dictionary. Adding more functions in the dictionary increases its redundancy, and thus, decreases the capacity of MP at finding the sparsest approximations. But having an *a priori* knowledge about the fact that some of the basis functions of the dictionary are more suitable for edge representation, and some of them are more suitable for textures allows to perform a modified MP search, that takes this information into account in the choice of the basis functions. It has been proved in [38, 39] that, if the *a priori* is properly chosen, it can help MP find the sparsest solution in some cases where, without the *a priori*, it would fail.

This section introduces a new set of basis functions that are more efficient at texture representation than the anisotropic refinement atoms. It presents some results of image approximations using this new bigger dictionary, including atoms capable of representing textures, anisotropic refinement atoms and the low-pass Gaussian atoms. Then, presents the computation of the weights in order to help MP finding a sparser approximation.

Intuitively, MP chooses at every iteration the basis function of the dictionary that is more similar to the signal (maximum projection). In some cases, this leads to suboptimal approximations. Intuitively, if an *a priori* knowledge about which basis functions of the dictionary are more suitable for a given approximation, one can help MP by penalizing the functions that are not supposed to be in the optimal approximation. This penalization can be just an α factor (with $0 < \alpha_\gamma \leq 1$) that multiplies the scalar product. Then, MP does not any more choose the basis function that has the most energetic projection at every iteration, but the basis function that has the most energetic weighted projection (where the weight is atom dependent). Analytically, it can be seen as computing:

$$\mathbf{g}_{\gamma_m} = \arg \max_{\gamma} \alpha_\gamma |\langle \mathbf{R}^m \mathbf{s}, \mathbf{g}_\gamma \rangle|, \quad (6.33)$$

where γ is the index that defines the basis function of the dictionary. As it has been mentioned in Chapter 4, in [182, 184], Temlyakov demonstrated that such a weighted scheme, with some constraint on the weights, converges. Nevertheless, the intuitive fact that if the weights are properly chosen, they may help MP to converge did not appear in these studies. Lately, it has been demonstrated in [38] that if the weights are properly chosen, this weighted MP does not only converge but it can find the sparsest approximation in some cases where standard MP does not find it, due to the excessive redundancy of the dictionary. Thus, it seems logical to use this *a priori* knowledge, if it exists.

Images seem a good candidate to have several families of basis functions in the dictionary, because they have different structures with different characteristics. An image has mainly three kinds of differentiate structures: smooth parts, textures and edges. Smooth parts are efficiently represented by smooth functions and edges are efficiently represented by anisotropic refinement atoms. A dictionary to represent textures has to be found, that takes profit of the main features of textures: pattern repetition, localization in space, etc. A weighted MP scheme that has a very redundant dictionary formed by the union of three basis function families is presented here.

6.8.1 Edge dictionary

The edge dictionary of this scheme is the same that has been introduced in Section 6.2, thus, formed by anisotropic refinement atoms. The same sampling of the parameters is the same that has been explained before (one pixel translation, octave based scaling and uniform angle quantization).

6.8.2 Texture dictionary

Textures are difficult to model, and a lot of work has been done to better understand their nature [97, 121, 127, 176, 196]. The main characteristic of textures is that they form a pattern. This pattern may be very regular, such as a brick wall or a striped shirt, or more random, as it is the case with the fur of *Baboon*, the hat of *Lenna* or the grass of *Cameraman* (see Figure 6.6). The dictionary that is more appropriate for sparsely representing sparsely should be localized in space (as textures are, in most of the images, bounded to a region of this image). The basis functions used to represent textures should also be localized in frequency, in order to have the space-localization properties described in the previous sections and be able to use the FS-MP algorithm. It is well known that the Gaussian envelope gives a good compromise in frequency and space localization, and this kind of envelope has proved to give good results with contours and smooth regions. Thus, it is interesting to keep on using it. The dictionary of basis functions adapted to frequencies will be just Gabor atoms (modulated Gaussians), which have indeed been already used for texture segmentation [97]:

$$g_\gamma(x, y) = \sqrt{2}K e^{-(x^2+y^2)} e^{i(\omega_x x + \omega_y y)}. \quad (6.34)$$

where K is a normalizing constant. As images are real signals, just the real part of $g_\gamma(x, y)$ is considered:

$$g_\gamma(x, y) = \sqrt{2}K e^{-(x^2+y^2)} \cos(\omega_x x), \quad (6.35)$$

where $\gamma = (\mathbf{p}, a, \theta, \omega_x)$ defines the set of parameters that apply to the basis function, similarly to the AR or the Gaussian dictionary. The use of these parameters to define the basis function follows the same logic than the case of AR atoms. As the sense of the frequency structure can be at any orientation, these kind of atoms have an angle parameters that orients them parallel to the structure of the signal to represent. The size of the structure can vary from signal to signal, and thus, a dilation parameter has to be applied to this kind of atoms. Of course, translations all over the pixels of the image are needed as well. In addition, several modulating frequencies have to be taken into account, in order to be able to capture structures that have quick variations and textures with wide structures. The minimum modulation frequency is taken such that the atoms have more than two local minima in them. This is done because, as it is explained in the next subsection, the edge dictionary is formed by anisotropic refinement atoms, that have two local minima in them, and thus, if this kind of structure is needed, it can be efficiently represented by AR atoms.

6.8.3 Smooth parts dictionary

As in the case of the dictionary formed by anisotropic refinement atoms, the dictionary formed by Gabor atoms and anisotropic refinement atoms is not capable of representing the low frequencies of the signal. The Gaussian subdictionary (see Section 6.6) is added, to capture these frequencies.

6.8.4 Comments about the final dictionary

The three dictionaries presented in this section are put together in a macro-dictionary, which has basis functions that are efficient for edge representation, basis functions efficient for smooth zone representation and basis functions for texture representation. The final dictionary is highly redundant,

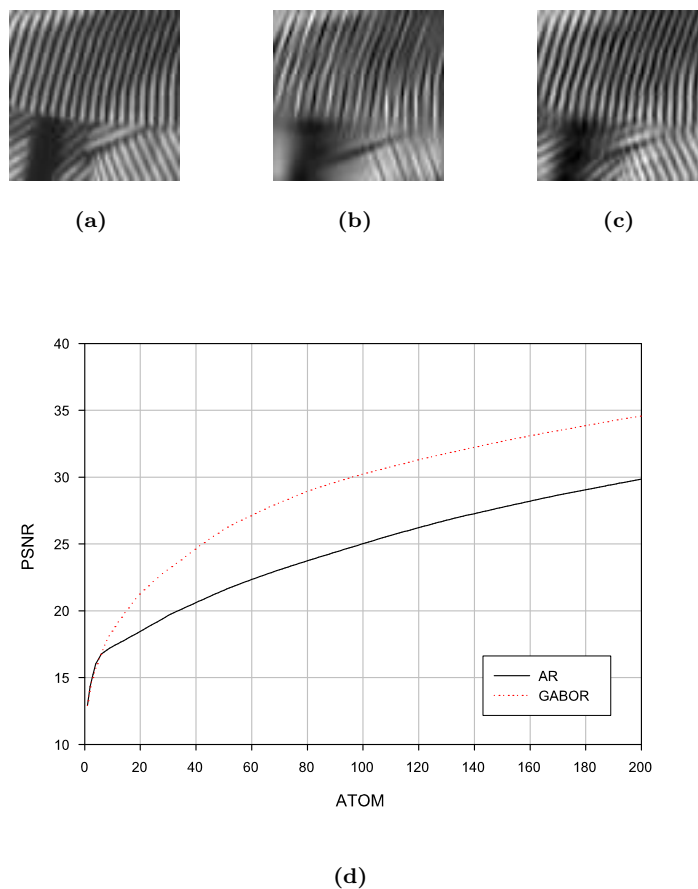


Figure 6.19: Example of texture coding with different dictionaries [19]. Subfigure (a) corresponds to a detail of Barbara and (b) and (c) to the MP coding with 50 AR atoms and Gabor atoms respectively. It can be seen both through the visual comparison and through the PSNR plot that Gabor atoms capture more efficiently the textures than AR atoms.

formed by three subdictionaries. MP is normally not capable of finding the sparsest representation in such a highly redundant dictionary, and thus, some strategy to help it has to be found. As one has three subdictionaries, each one adapted to a different image part, intuitively one can think about forcing MP use every dictionary in the image part it has been designed for. This can be done by imposing some weighting masks, which will improve MP convergence [38, 39]. The question is now how these weighting masks can be computed and used.

6.8.5 Computation of the weights

The computation of the weights is the most critical part of the Weighted MP (WMP) representation scheme. If the weights are properly computed, these cause WMP to perform better than the standard MP. On the contrary, a bad computation of the weights causes a loss of approximation quality. One can think of several methods in order to compute the weights. Probably the most straightforward is to perform edge and texture detection, and to weight with higher values the texture regions for the subdictionary of textures, and the edge regions for the subdictionary of edges. Intuitively, as the dictionaries have been designed to be respectively performing for edges and for textures, this

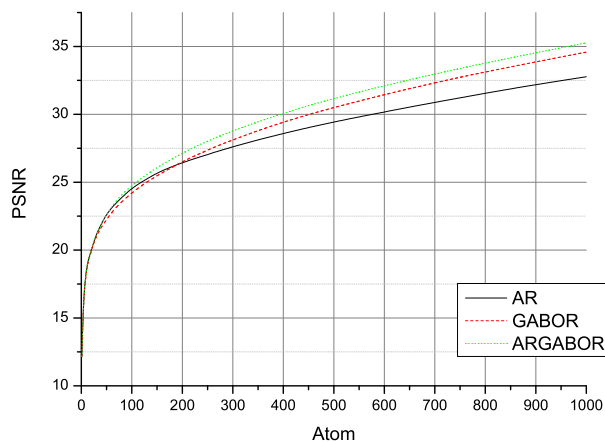


Figure 6.20: PSNR for MP approximations using various dictionaries types over Barbara image. It is shown that the combined use of Anisotropic Refinement and Gabor dictionaries (AR GABOR) leads to better PSNR ratios than the use of single dictionaries (AR or GABOR) because it catches better the structures of the image.

should improve the approximation properties of the MP algorithm.

In this section, a definition of the concept of the weight mask is given, in order to better understand its meaning and its effect in the decomposition. Its computation is as well presented, both for the texture dictionary and for the edge dictionary. Finally, some examples of the masks are given.

Definition

A probability mask or weight mask can be defined as a matrix

$$\mathbf{W} \in [0, 1] \quad (6.36)$$

where each element has a real positive value limited between 0 and 1. This value indicates the probability of a certain element of the dictionary to take part in the signal representation.

For the concrete case of the dictionary presented here, formed by AR atoms, Gaussian atoms and Gabor atoms, and in order to be able to keep on using the same full search scheme that has been presented in this chapter, the weight mask will be position based. The Gaussian atoms, as they are used only in the first iterations of the MP algorithm, will not have any weight smaller than one. The Gabor and the Anisotropic Refinement atoms, on the contrary, have some masks that influence in which positions the MP algorithm uses them. Every basis function in the Gabor subdictionary has the same position weighting mask, and, similarly, every basis function in the AR dictionary has a common position weighting mask. Thus, the weighting masks tells, in this concrete case, where it is more probable to use a Gabor atom (where textured zones are located in the image) and where it is more probable to use AR atoms (contours). In order to compute the probability masks of edges and of textures (respectively, of AR atoms and of Gabor atoms), the behavior of AR atoms and Gabor atoms when approximating a signal has to be taken into account.

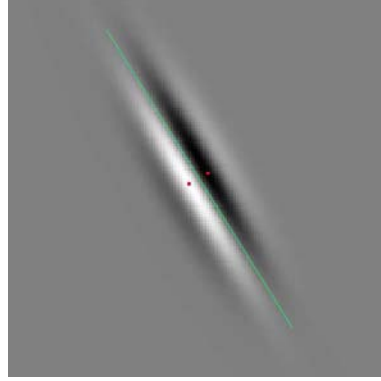


Figure 6.21: Edge representation by two Anisotropic Refinement atoms. When MP represents an edge, assuming a simple model of two atoms, the result looks like this. The green line represents the edge and the two red spots the center of the two atoms. This figure shows that to shape a step edge, the atoms are not on the edge itself but near it. Nevertheless, not all the edges are step edges, and not all of them require of two AR atoms to be approximated.

Edge position mask

AR atoms are appropriate for edge approximation, and thus, the positions in the image where they are supposed to be placed is near the image contours. The weights for AR atoms should be, then, of value one around contours, and of value smaller than one everywhere else. Some more sophisticated mask for AR atoms could be built, by estimating the orientation of contours, and performing different masks for the different angles. This, though, does not bring a great improvement in the choice, of the basis functions because this kind of atoms are very selective to directions. Thus, only one mask, common for all the functions in the AR subdictionary, that helps the MP place AR atoms on edges and not on textures is used.

In order to create the edge mask, the way AR atoms represent edges has to be taken into account. To do this, an edge is modeled by its most simplified version, a Heaviside function:

$$\mathcal{I}(x, y) = \begin{cases} I_{\min}, & \text{if } ax + y < 0 \\ I_{\max}, & \text{if } ax + y > 0 \end{cases} . \quad (6.37)$$

The representation of a Heaviside function with AR atoms is studied, and results are extrapolated to understand the exact behavior of this kind of atoms toward edges, as shown in Figure 6.21.

AR atoms are naturally placed with their oscillatory axis perpendicular to edges. Knowing this, it is possible to simplify the study of the behavior of this kind of atoms with respect to edges by taking only a 1D signal that is the AR atoms oscillatory part. An edge is then represented by a 1D Heaviside function, and only the oscillatory part of AR atoms is kept. The 1D Heaviside function naturally becomes:

$$\mathcal{I}(x) = \begin{cases} I_{\min}, & \text{if } x < 0 \\ I_{\max}, & \text{if } x > 0 \end{cases} , \quad (6.38)$$

and the 1D version of the AR atom is:

$$g_{\gamma}(x) = \left(4 \left(\frac{x - p_x}{a_x} \right)^2 - 2 \right) e^{-\left(\frac{x - p_x}{a_x} \right)^2}, \quad (6.39)$$

with $\gamma = [p_x, a_x]$ the function index, that has as parameters translation and scale. If a two term

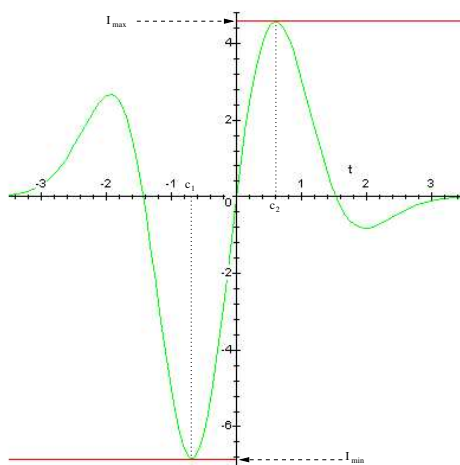


Figure 6.22: Edge Model. The red plot models a step edge, while the green represents the simplest representation of such an edge by taking a linear combination of two Anisotropic Refinement atoms.

edge approximation is wished, the expression of the edge approximation becomes:

$$\tilde{I}(x) = c_1 \cdot g_{[p_{x_1}, a_{x_1}]}(x) + c_2 \cdot g_{[p_{x_2}, a_{x_2}]}(x) \quad (6.40)$$

where p_{x_1} are p_{x_2} the center of each atom, a_{x_1} and a_{x_2} their respective scaling factors and c_1 and c_2 the coefficients, that depend on the amplitude of the edge.

In order to know where atoms are placed around edges, the distance p_{x_1} that maximizes the scalar product between the atom and the edge for a given scale a_{x_1} has to be computed. If this is done with the Heaviside function, it can be seen that the center of the atom is at a distance d from the edge proportional to its scaling parameter:

$$d = \frac{\sqrt{2}}{2} s_x. \quad (6.41)$$

This shows that, for this kind of edges, the edge position probability mask needs to be one not only on the edge himself, but inside two regions parallel to the edge and at each side (see Figure 6.23), at starting and ending distances:

$$d = \left[\frac{\sqrt{2}}{2} s_{x_{\min}}, \frac{\sqrt{2}}{2} s_{x_{\max}} \right]. \quad (6.42)$$

Nevertheless, edges are not always symmetric, and cannot always be modeled by a Heaviside function (most of natural edges are not a step function but a smoother). Thus, there is a need to put high probability of using AR atoms on the edges as well.

To build the edge weight mask, the first thing to be done is, of course, to detect edges. Standard edge detectors [18, 114, 122, 125, 133, 145] give a positive output on edges, and zero or very small values everywhere else. It can be assumed that the output of the edge detector is one on edges, and zero everywhere else (if this is not the case, a simple thresholding operation can give it). Some post-processing has to be done in order have a wider zone, that goes from the edge to a distance $\frac{\sqrt{2}}{2} s_{x_{\max}}$ at both, with this high value. This wider area around the peak can be easily obtained

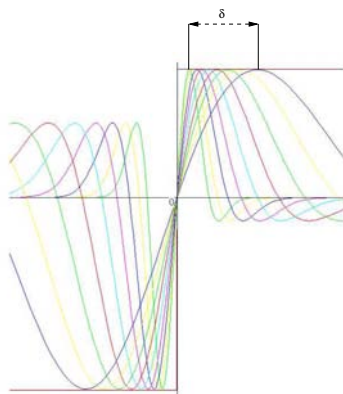


Figure 6.23: Atom placement area δ .

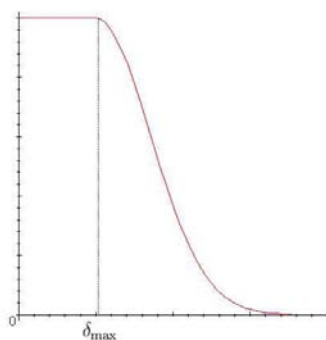


Figure 6.24: Impulsional response of the edge model mask linear filter

through linear filtering [19]. It can be useful, too, that the output of this filter does not present discontinuities. A possible design of such a filter is:

$$\varphi(r, \theta) = \begin{cases} 1, & \text{if } r < d_{\max} \\ e^{-\left(\frac{r-d_{\max}}{\sigma}\right)^2}, & \text{if } r > d_{\max} \end{cases}, \quad (6.43)$$

where the filter is expressed in polar coordinates and σ is the parameter that defines the smoothness of the decay of the weight mask values. Figure 6.24 shows the right hand output of such a filter with respect to the radius. As it is independent on the angle, the filter is isotropic.

Once this filter has been applied to the output of the edge detector, the resulting weighting mask is an image that depicts the edges of the image and that has a soft transition among edgy and non-edgy parts. This soft transition is useful, in order to allow MP to correct the oscillatory effects that may result from the use of AR atoms.

A last point to be taken into account for the edge mask is that not all the edges have the same strength. Most edge detectors give, before thresholding, an output that is directly dependent on the strength of the edge. As normally stronger structures are more important for the visual impression, it can be useful to take this into account in the edge mask. Nevertheless, MP usually takes the strength of the edge into account intrinsically, because stronger edges usually give stronger scalar products.

Texture position mask

Gabor atoms mask is easier to compute than the edge mask, because Gabor atoms have to be placed just inside textured parts. Thus, a texture detector tells directly in which zones Gabor atoms have to be placed and in which positions of the image the Gabor atoms mask has to have values of one. All the zones that do not present textures, or that present very weak textured zones, have values smaller than one in the Gabor atom mask. The computation of the Gabor atoms mask is performed just by applying a texture detector, and setting to one the zones where a texture has been detected:

$$\mathbf{W}_T(x, y) = T[I(x, y)], \quad (6.44)$$

where T is the texture detector and I is the original image. As a certain regularity is wished in the zone detection, it may be appropriate to perform a low-pass filtering of the output of the texture detector.

Feedback between textured and edgy zones

Edge detectors and texture detectors do not perform always properly. Generally, edge detectors give a positive output in textured zones, because they see a texture as a mix of edges. Thus, it can be useful to use the output of the texture detector as a feedback for the edge mask. The zones that have positively been detected as textures can be set to smaller values in the edge detector output, before the convolution of the edge detector output with the linear filter that models the position or AR atoms around edges. The output of this improved edge detection can be used to eliminate from the texture detector output zones that are mostly edgy, and that have anyway a positive output in the texture detector. This simple operation allows for better weighting masks for edge and texture position.

Mask impact control

The computation of the weight masks as explained in Section 6.8.5 gives output values that are in the interval $[0, 1]$. This scheme may be, in some cases, too penalizing (some AR atoms or some Gabor atoms may be needed for zones that have not been detected as edgy or as textured). In order to control the effect of the weighting mask, one can just impose that the minimum value of this mask is higher than a positive constant $\alpha < 1$. Of course, the higher α , the lower the effect of the weighting mask is in the choice of the basis functions, and thus, the lower the selectivity of the mask. Lower α implies that more basis functions are rejected, and thus, that MP is able to find the sparsest solution with more redundant dictionaries. Having a very low α implies, too, that in case the weighting mask is not accurate, more approximation accuracy is lost. As edge and texture detectors are not one hundred per cent accurate, it is worth allowing MP to choose some functions that have not been detected as being in the sparsest solution, by putting α big enough. Still, it has to be small enough so that some help is given to the choosing algorithm. If α is chosen very big, it is useless to have these weighting masks.

There are several ways to control the impact of the weighting masks: to shrink their values from α to 1 by simply scaling them and adding a constant, to waterfill or watershed [149, 200] the mask. The first will keep the errors the edge or texture detector may have done (errors that are normally in the low energy structures). Performing a waterfilling has the advantage of smoothing the mask while reducing the possible errors (low energy structures, see Figure 6.25). It seems thus appropriate to perform a waterfilling of the mask before applying it to the MP algorithm. The waterfilled version

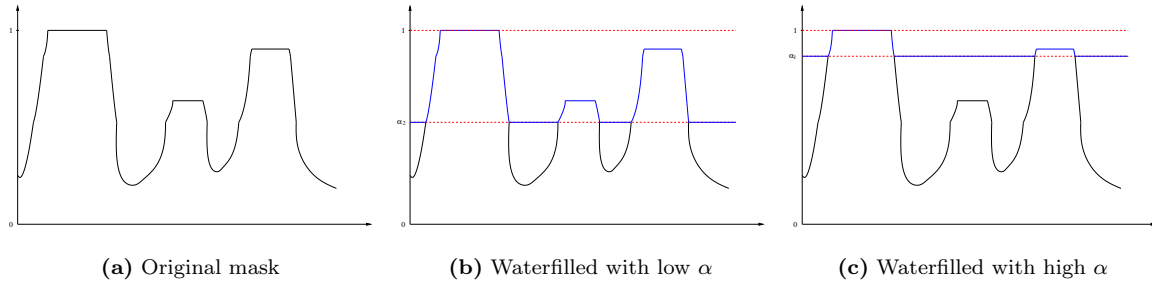


Figure 6.25: Waterfilling. The α parameter allows to control the influence of the masks into the pondering process. In (a), the original a 1D-Mask. In (b), its waterfilled result with a high value of α and, in (c), with a lower value. Note that in (c) begins to appear a weak probability region (lower peak).

of the mask can be expressed as:

$$\mathbf{W}_\alpha(x, y) = \begin{cases} \mathbf{W}(x, y), & \text{if } \mathbf{W}(x, y) > \alpha \\ \alpha, & \text{otherwise} \end{cases}, \quad (6.45)$$

where $\mathbf{W}_\alpha(x, y)$ is the mask with impact control and $\mathbf{W}(x, y)$ the edge or texture mask.

6.8.6 Weighted MP image representation

With the position probability masks presented (edge and textures), the weighted MP can be implemented as shown in Figure 6.26: every convolution of a centered basis function with the image is multiplied by the normalizing mask (as done in Section 6.4) *and* the weighting mask of the corresponding dictionary (AR atoms or Gabor atoms, the Gaussian atoms do not have any mask).

Of course, the masks for the AR atoms and for the Gabor atoms have to be computed once at the beginning of the image representation process, and kept in memory. Nevertheless, as only one position mask per dictionary is taken into account, this does not represent a significant increase in memory usage or computational time, specially because the weight mask and the normalization mask can be integrated in a single matrix for every centered version of a basis function of the Gabor or the AR dictionary.

Notice that the AR atom mask (the edge region mask) has been designed to be suitable for all the atoms of the AR subdictionary, from the tiniest scale to the larger one. A more elaborated version of this mask could take into account the different scales, in order to adapt the width of the flat part of the filter to the needs of every scale. This can help the MP algorithm avoid correcting edges that have already been selected, and concentrate on the representation of the global sketchy image before starting correcting the approximation artifacts. This “scale-personalized” edgy mask can be as well integrated on the normalization mask, as there is a normalization mask for every orientation, x and y scale combination.

The concept of using masks to enhance areas with more visual information was already proposed in [207]. This scheme differs from foveated image representation by the fact that the masks here are not use to enhance any visual information, but just to help MP find the sparsest solution. The result, though, is quite similar: the masks cause the algorithm to prefer highly textured or edgy zones, which coincide with areas with rich visual information.

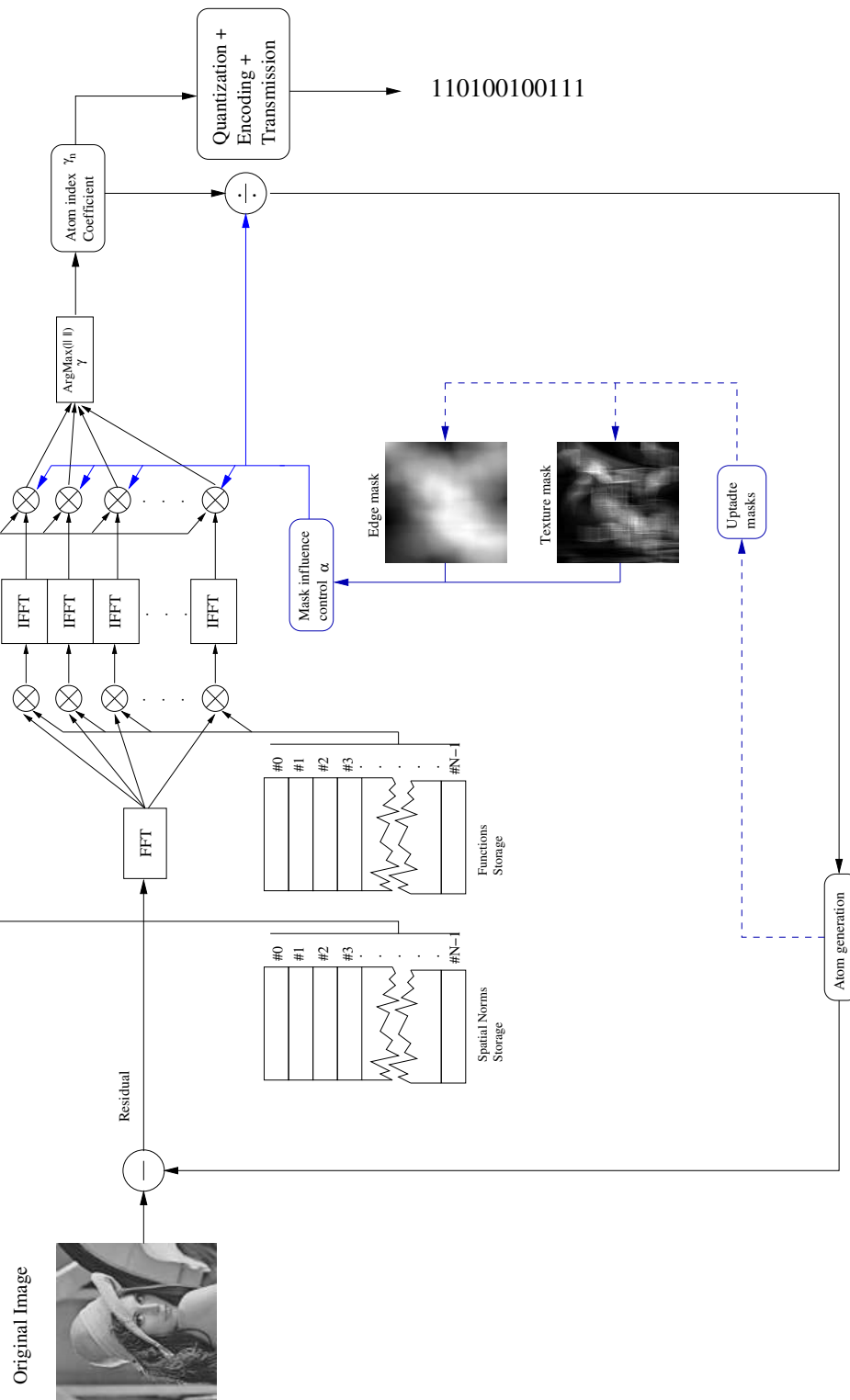


Figure 6.26: Structure of the Hybrid MP coder. It can be seen the Hybrid MP coder works similarly to the FS-MP coder, but with a pondering of the scalar product energy when in the choice of the basis functions.

6.9 Color images: multichannel MP

6.9.1 Vector MP

When applying MP for color images, the signal \mathbf{s} is no more a simple image, but a vector of images. Color images have typically three channels. The color image representation that most image capturing devices give represented in the RGB color space (Red, Green, Blue). This color space is quite similar to the capturing system of the human eye in day-light conditions, where there are three kinds of receptors, the Red cones, the Green cones and the Blue cones (see Chapter 2, Section 2.3). Of course, in the Human eye there are luminance receptors as well, but these are mostly used in poor light conditions.

Instead of performing independent iterations in each color channel, a vector search algorithm is used in the proposed color image encoder. This is equivalent to using a dictionary of P vector atoms of the form $\{\mathbf{g}_{\gamma_{\text{color}}} = [\mathbf{g}_{\gamma}, \mathbf{g}_{\gamma}, \mathbf{g}_{\gamma}]\}_{\gamma \in \Gamma}$. In practice though, each channel is evaluated with one single component of the vector atom, whose global energy is given by adding together its respective contribution in each channel. MP then naturally chooses the vector atom $\mathbf{g}_{\gamma_{\text{color}}}$, or equivalently the vector component \mathbf{g}_{γ} , with the highest energy. Hence the component of the dictionary chosen at each MP iteration satisfies:

$$\gamma_n = \arg \max_{\gamma_n} \sqrt{\langle \mathbf{R}^n \mathbf{s}_1, \mathbf{g}_{\gamma_n} \rangle^2 + \langle \mathbf{R}^n \mathbf{s}_2, \mathbf{g}_{\gamma_n} \rangle^2 + \langle \mathbf{R}^n \mathbf{s}_3, \mathbf{g}_{\gamma_n} \rangle^2}, \quad (6.46)$$

where $\mathbf{R}^n \mathbf{s}_i$, $i = 1, 2, 3$ represents the signal residual in each of the color channels, instead of:

$$|\langle \mathbf{R}^n \mathbf{s}, \mathbf{g}_{\gamma_n} \rangle| = \arg \max_{\mathcal{D}} |\langle \mathbf{R}^n \mathbf{s}, \mathbf{g}_{\gamma} \rangle|, \quad (6.47)$$

the MP search criteria for single channel MP.

Note that the vector MP scheme presented in the scope of this thesis is slightly different than the algorithm introduced in [116], where the sup norm of all projections is maximized :

$$\max_i \sup_{\mathcal{D}} |\langle \mathbf{R}^n \mathbf{s}_i, \mathbf{g}_{\gamma_n} \rangle|. \quad (6.48)$$

The use of the quadratic sum is not mandatory: one could think of using any other vector norm, depending on the application. For example, for certain applications, it can be useful to use the ℓ^1 norm instead of the ℓ^2 one.

All signal components are then jointly approximated through an expansion of the form :

$$\mathbf{s}_i = \sum_{n=0}^{+\infty} \langle \mathbf{R}^n \mathbf{s}_i, \mathbf{g}_{\gamma_n} \rangle \mathbf{g}_{\gamma_n}, \quad \forall i = 1, 2, 3. \quad (6.49)$$

Note that channel energy is conserved, and that the following Parseval-like equality is verified :

$$\|\mathbf{s}_i\|^2 = \sum_{n=0}^{+\infty} |\langle \mathbf{R}^n \mathbf{s}_i, \mathbf{g}_{\gamma_n} \rangle|^2, \quad \forall i = 1, 2, 3. \quad (6.50)$$

The search for the atom with the highest global energy necessitates the computation of the three scalar products $c_n^i = \langle \mathbf{R}^n \mathbf{s}_i, \mathbf{g}_{\gamma_n} \rangle$, $i = 1, 2, 3$, for each atom \mathbf{g}_{γ_n} , and for each iteration of the MP expansion. The number of scalar products can be reduced by first identifying the color channel with the highest residual energy, and then performing the atom search in this channel only. Once the best atom has been identified, its contribution in the other two channels is also computed and encoded. The reduced complexity algorithm performs in a suboptimal way compared to the maximization of the global energy, but in most of the cases the quality of the approximation does only suffer a minimal penalty, as can be seen in Figure 6.29(b), which presents an example of MP performed in the most energetic channel.

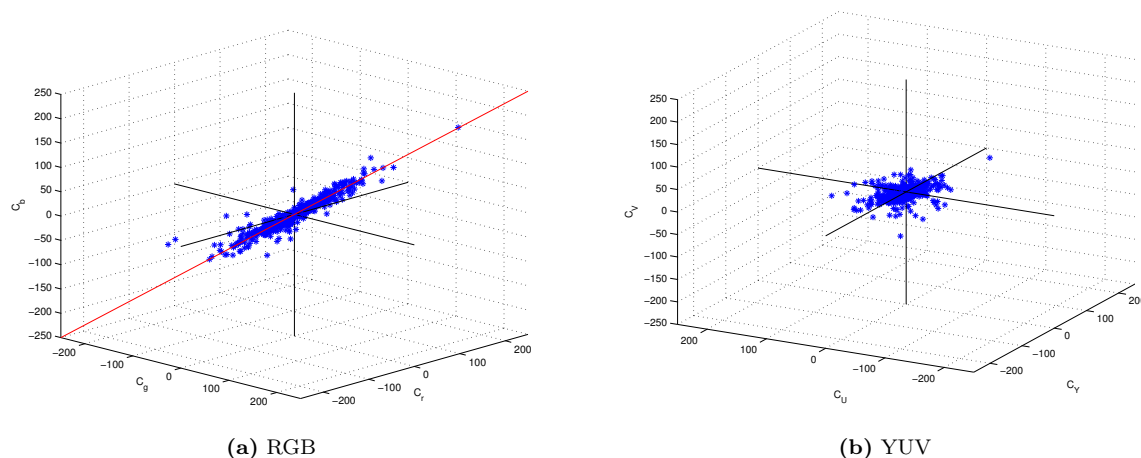


Figure 6.27: Distribution of the MP coefficients when MP is performed in the RGB color space (left), with the diagonal of the cube shown in red, and in the YUV color space (right).

6.9.2 Color space choice

The choice of the color space has to be adapted to the proposed image representation strategy, and minimize the distortion introduced to the image due to the representation scheme itself. As the proposed image representation scheme tries to achieve sparsity by taking profit from the fact that color channels have a high correlation among them, it is logical to choose the color space that presents more correlation among channels.

In order to reach a maximal correlation between the color channels, and thus the possibility to have high and similarly valued coefficients for the same basis function in the three channels, the RGB color space seems a natural choice. RGB color space is the only one that has not performed any subtraction operation, or any linear combination in order to reduce redundancy among channels. RGB color space presents, in most of the natural images, the same edge at the same position of the image for the three channels, just with some differences on the importance of the edge. It has to be seen, now, if this space really presents coefficients with similar energies at the three channels.

This fact is proved through the following simple experiment. The coefficients $[c_n^1, c_n^2, c_n^3]$ of the MP decomposition can be represented in a cube, where the three axes respectively corresponds to the projections of the basis functions in the red, green and blue components of the image (see Figure 6.27(a)). It can be seen that, when using RGB color space projection, the MP coefficients are interestingly distributed along the diagonal of the color cube, or equivalently that the contribution of MP atoms is very similar in the three color channels. On the contrary, the distribution of MP coefficients resulting from the image decomposition in the YUV color space, does not seem to present an obvious structure (see Figure 6.27(b)), and having large projection values of a dictionary basis function in one channel does not mean having large values in the other two channels. Thus, RGB color spaces seem the most appropriate to take advantage of the redundancy among channels and to provide flexible signal representations.

Most of the image coding techniques use color spaces such as YUV, LAB or CrCbCg. These color spaces have less redundancy among channels than RGB. For example, YUV has all the luminance information in the Y channel, and the U and V channels have less information. LAB and CrCbCg color spaces have the drawback that they have some user defined parameters, not standardized for



Figure 6.28: Original Japanese woman image

all the displays. They do not present the same amount of redundancy among channels as RGB does. The techniques which use the YUV color space generally take profit from the fact that the human eye is less sensitive to color than luminance localization in images, and hence downsample the U and V channels in order to reduce the amount of data. Such a downsampling is not helpful anymore in the context of the proposed MP coder, since the same function is used for the three color channels, in order to limit the coding cost of atom indexes. Since, in addition, the use of the same function in Y, U and V channels may induce color distortion, RGB becomes clearly the preferred color space for the MP coder.

6.10 Conclusions

This chapter has presented a FS-MP implementation, which uses the Fourier Transform in order to speed the computation of the scalar product between the functions in the dictionary and the signal to represent. It has shown that the approximation properties of this FS-MP algorithm using anisotropic refinement atoms are very good, compared to standard orthonormal basis. The same chapter has introduced, as well, a new low-band dictionary, that can be used together with the Anisotropic Refinement Atoms dictionary in order to give fully flexible image representations, allowing intrinsic geometric operations, such as reconstruction of the images at different scales, translations or rotations.

This FS-MP Algorithm has then been used to explore a new image approximation scheme, that combined three dictionaries (one for low-frequencies, one for contours and one for textures). As the macro-dictionary formed by these three subdictionaries becomes really complex, some weighting masks have been presented, that help choosing, for different image structures, the good subdictionary of functions.

Finally, a novel way to adapt the MP image approximation to color images has been presented.



(a) YUV most energetic



(b) RGB most energetic

Figure 6.29: (a) Japanese woman coded with MP using the most energetic channel search strategy in YUV color space and (b) in RGB color space.

This color image representation has the particularity that, instead of trying to decorrelate the three color channels and code them independently, takes advantage of this correlation. The Color MP image representation keeps the same capability to perform geometrical image transformation in the transformed domain than the gray-scale MP image approximation scheme, as the same dictionary of functions has been used.

It has been seen, thus, in this chapter, that MP allows for sparser image approximations than existing techniques. It has to be seen, now, if this gain in sparsity gives a gain in coding performances as well. This fact, together with the study of the coding scheme to follow with such a signal approximation, is explored in the following chapter.

Flexible image coding with Matching Pursuit

7.1 Introduction

Matching Pursuit with a dictionary formed by anisotropic refinement atoms and Gaussians has proved to provide good approximations with a small number of terms. It seems logical that such a scheme can give good very low bit-rate compression. Nevertheless, this is not straightforward, due to the fact that, even though the number of terms is very small, the function index is very complex to code, because the dictionary size is very large. On the other hand, the quantity of information that this index brings is very large. In order to use this scheme for image coding, a detailed study about the optimal MP coefficients quantization and the function index coding has to be performed.

This chapter examines the coding performances of a Matching Pursuit based on anisotropic refinement atoms and Gaussians. A detailed study about the best strategy to code the atom indices and to quantize and code the coefficients is presented. Finally, a flexible coder based on Matching Pursuit, suitable for very low bit-rate applications, is introduced. This coder gives a scalable stream, that allows for geometrical transformations in the compressed domain. The aim of this coder is to be used in asymmetric applications, such as wireless networks. The requirements it has to accomplish are:

- to be fully rate-scalable. Fully scalable means that the bit-stream can be truncated at any point without implying impairing the reconstruction of the image at the receiver. In addition, the loss of quality due to truncation must remain small. In order to check if the loss of quality remains small, the truncated stream images has to be compared to a reconstructed image coded directly at the truncated rate.
- to allow flexible geometric manipulations in the coded domain, in order to be able to reconstruct the image at any size.

The requirement of being able to reconstruct the image at any size is needed due to the large variety

of end-user terminals existing today, specially for mobile communications, which causes a large variety of end-user terminals, receptor bandwidth capacities and scree sizes.

This chapter is organized as follows: Section 7.2 introduces the function indexation of the MP coder. It discusses the complexity of this function indexation, and it presents different schemes that can reduce the entropy of this indexes. Section 7.3 introduces a quantization scheme built on theoretical results derived in [72, 75] and an enhanced dequantization derived from empirical observation of the evolution of the MP coefficients through the iteration number. Section 7.4 presents an image coding scheme that use MP expansions and the practical implementation of the quantization scheme presented in Section 7.3.2. Section 7.5 shows the coding performances of such a scheme, while Section 7.6 evidences the flexibility such a coder gives. The coding scheme presented until this point is based on the theoretical upper-bound of the MP coefficients, but it does not take into account its real distribution. Section 7.7 states how to extend this coding scheme to color image compression with Matching Pursuit, following a similar coding scheme than the one used for gray-scale images. Finally, Section 7.8 gives some conclusions

7.2 Reduction of the entropy of the function index

The dictionary of functions used to achieve sparse image representations is very large, and if the indexation of the function is performed just by one single index, this index takes very large values and suffers no repetitions along a single image coding. The consequence is that the entropy of this index, and thus the number of bits needed to encode it, becomes too large. To solve it, a smarter indexation of the basis functions has to be performed. This section analyzes the strategy to follow in order to reduce the number of bits needed to represent the basis function indexation.

As the basis function of this dictionary are created by applying affine transformations to a single basis function, it sounds logical to use as function indexation the parameters of this affine transformation. Doing this, some parameters of this index will be repeated from atom to atom, and thus, some indexation bits can be spared. Moreover, the fact of sending the basis function affine transformation parameters allows to reconstruct basis functions that were not in the initial dictionary (the basis function do not need to be stored, but can be built on the fly both at the encoder and the decoder), and to perform affine image transformations directly in the coded domain. In addition, this split index coding allows to save some bits for the functions belonging to the Gaussian subdictionary. The Gaussian atoms can be completely defined by three parameters: position x , position y and dilation. The anisotropic refinement atoms need two more parameters to be completely defined. They need, apart from the same three parameters of Gaussian atoms, two extra indices: an extra scaling factor (because they have anisotropic scaling) and the rotation parameter. To have the whole dictionary indexed, an extra parameter is needed, in order to perform the subdictionary choice. Thus, the function index is of variable size: it has four integers if the basis function in use is a Gaussian (kind of function, x position, y position and dilation) and six if the basis function in use is an AR atom (kind of function, x position, y position, x scaling, y scaling and rotation). Of course, for both the Gaussian subdictionary and the AR atoms subdictionary, the projection value of the basis function has to be sent together with the basis function index himself.

Several strategies can be taken in order to reduce the entropy of the indices. In addition, the function index should be analyzed, in order to better understand if it is possible to reduce the amount of bits needed to transmit them. The main constraint that exists, in the reduction of the entropy of the MP parameters, is the constraint of the full rate-scalability, which requires to send the MP coefficients sorted by decreasing projection energy. Strategies such as ordering the MP coefficients by position and sending them by differential position coding [135, 136] cause a loss of

this rate-scalability. Nevertheless, in the case of natural images, this differential position coding does not bring a significant reduction of the entropy of positions, due to the fact that the number of terms in use is much smaller than the number of positions in an image, and thus, the range reduction due to differential position coding is not significant. Several other schemes have been tested, in order to reduce the entropy of the positions, such as quadtree position coding, but with none of little improvement with respect to sending the raw positions: due to the (almost) uniform distribution of the atom positions in the image and the small (compared to the size of the image) number of atoms that are used for an image approximation, the improvements provided by these schemes are not significant, and in addition they cause a loss of rate-scalability. The position indexing, thus, has been sent without any modification.

The scaling parameters differ from the Gaussian atoms and the Anisotropic Refinement atoms. The Gaussian atoms dilation parameter starts at a size that is dependent on the image size, but that is fixed for a given image size. As both the receptor and the receiver know this amount, the Gaussian dilation parameter is sent as $a - a_{\min}$, a_{\min} being the minimum Gaussian scaling parameter allowed. The range of this parameter is then from 0 to $a_{\max} - a_{\min}$.

In the case of anisotropic refinement atoms, the scaling parameters are sent as follows: the y scale is always bigger than the x scale, in order to have more efficiency in the representation of contours (the oscillatory part has always to be smaller than the smooth part). The x scale starts at one, but the parameter that is sent is a_x , with $2^{\frac{a_x}{NN}}$ (where NN defines the number of scales per octave and is defined in Appendix B), which starts at zero. The scaling parameters have been discretized in an octave based distribution, in order to keep the dictionary density approximately constant (see Appendix B for a more detailed explanation). The y scale is similar to the x scale, but the minimum y scale allowed depends on the x scale:

$$a_x \in [0, \lceil NN \times \log_2 size_{im} \rceil] \quad (7.1)$$

$$a_y \in [a_x, \lceil NN \times \log_2 size_{im} \rceil]. \quad (7.2)$$

In order to reduce the entropy of the y scale parameter, then, the coder sends to the receiver $a_y - a_x$ instead of sending a_y .

The angle parameter is, similarly to the positions: its distribution is quite uniform, and highly depends on the image. The angle parameter goes from zero to the allowed number of angles minus one. This allowed number of angles is normally 18. As the probability distribution of this angle parameter over a set of numerous natural images is approximately uniform, it does not allow an efficient encoding.

The main conclusions of the studies performed to reduce the entropy of the parameters is that the atom indexes are quite aleatory, and highly depend on the image. Thus, few can be done to reduce the number of bits needed to send this atoms. In order to reduce the number of bits needed for every coefficient and atom index for the MP scheme used here is, then, necessary to concentrate in the coefficient quantization.

7.3 *A posteriori* MP coefficients quantization

The previous section has stated the fact that the reduction of the number of bits needed to send the atom parameters can be mainly achieved through the proper quantization of the value of the projection of the basis function. Atom indices seem too complex and random to allow for a smart coding strategy. The quantization of these projections has then to be carefully designed, in order to provide a good rate-distortion behavior.

Most Matching Pursuit schemes that appear in the literature use *in loop* quantization [135, 136, 137]. These in-loop schemes have the advantage of compensating, in the following iterations, the quantization error performed at a given iteration. The main disadvantage of such schemes is that, if one wished a different rate target, the MP decomposition has to be recomputed again. As the coder presented in the scope of this thesis is intended for asymmetric applications that can have variable bit-rate requirements, it has been considered that an *out-loop* quantization scheme (*a posteriori* quantization) is more helpful, because it allows to quantify each end-user stream at its rate-distortion target requirements. This possibility is specially useful for big images, because the computational complexity for MP approximating a big image remains high. Thus, the in-loop quantization schemes that appear in the literature cannot be used: an *a posteriori* quantization scheme seems more appropriate.

In order to understand the *a posteriori* quantization scheme used in the scope of this thesis, the upper-bound of the energy of the coefficients in the MP expansion is introduced. This upper-bound is the basis of the MP *a posteriori* quantization scheme introduced in [72, 75]. It will be seen that the quantizer takes profit from this upper-bound in order to reduce the amount of bits needed to send the coefficient value without an increase in quantization error.

In this section, first a review of the theoretical results given in [72, 75] is given. From this theoretical results, a practical implementation of the quantizer that is derived from this results is built, with a new rate allocation strategy that improves the results obtained in [75]. Finally, an original modification of improved reconstruction of the quantized values is implemented.

7.3.1 Redundancy-Driven *a posteriori* MP coefficients quantization

The convergence speed of Matching Pursuit corresponds to its ability to extract the maximum signal energy in a few iterations. In other words, it corresponds to the decay rate of the residue. Since the convergence speed directly depends on the dictionary, the decay rate of the residual energy can be bounded once the dictionary is known, even without *a priori* information about the input signal.

The approximation error decay rate in Matching Pursuit has been shown to be bounded by an exponential [29, 118]. In other words, the decay of the residue norm is faster than an exponential decay curve whose rate depends on the dictionary only. From [118], there exists a decay parameter $\tau > 0$ such that for all $N \geq 0$

$$\|\mathbf{R}^M \mathbf{s}\| \leq 2^{-\tau M} \|\mathbf{s}\|, \quad (7.3)$$

or, equivalently,

$$\|\mathbf{R}^{m+1} \mathbf{s}\| \leq 2^{-\tau} \|\mathbf{R}^m \mathbf{s}\| \quad \forall m. \quad (7.4)$$

The decay rate* can be written as [118]:

$$2^{-\tau} = (1 - \alpha^2 \beta^2)^{\frac{1}{2}}, \quad (7.5)$$

where β is the redundancy factor and $\alpha \in (0, 1]$ represents an optimality factor. This latter factor depends on the algorithm that, at each iteration, searches for the best atom in the dictionary. The optimality factor α is set to one when MP browses the complete dictionary at each iteration (as it is the case with the MP scheme presented in the scope of this thesis). The parameter β depends on the dictionary construction. It represents the ability of the dictionary functions to capture features of any input function f . The upper-bound on the coefficient norm is reached in the worst case where

*The decay parameter τ decreases when the size of the signal space increases. However, at the limit of infinite dimensional spaces, the convergence is not exponential any more [99].

the input function is the farthest from any dictionary vector. β is defined as the largest constant that satisfies:

$$\sup_{\gamma} |\langle \mathbf{s}, \mathbf{g}_{\gamma_m} \rangle| \geq \beta \|\mathbf{s}\|. \quad (7.6)$$

β can be interpreted as the cosine of the maximum angle between any direction in \mathcal{H} and the closest element of the dictionary, that we simply assume to be a L -dimensional subspace of \mathcal{H} in the finite dimensional case [119]. This parameter characterizes the redundancy of the dictionary and tends to one when the size of the complete dictionary increases. A geometrical formulation of β is provided in [74].

Notice that β and μ (redundancy factor introduced in Chapter 4) are not at all the same parameter. β is computed as one half of the cosine of the bigger angle among two neighboring functions in the dictionary, while μ is the cosine of the smaller angle among two neighboring functions in the dictionary. Thus, $\mu > \beta$. For dictionaries that are uniformly distributed in the function space, β and μ are highly related, but these two parameters can be very different if the dictionary is not uniformly distributed in the function space.

The purpose of the *a posteriori* quantization scheme basically lies in the context of asymmetric applications, and is thus appropriate for the purpose of the image coder presented in this thesis. Due to the encoding complexity, the Matching Pursuit stream is computed only once and then differently quantized to match various receiver requirements. An optimal coefficient quantization cannot be included in the Matching Pursuit decomposition in contrast to [81, 82, 136], since this would imply dedicated encoding for each end-user. While *a posteriori* quantization provides flexibility regarding to the coding rate, the associated distortion is however not compensated by further MP iterations, contrary to in-loop quantization. The quantization error propagation can nevertheless be kept small for sufficiently fine quantization. A graphical explanation about the difference between in-loop and out-loop quantization can be found in Figure 7.1.

Figure 7.2 shows that distortion resulting from *a posteriori* quantization are slightly higher than those of the in-loop quantization, depending on the quantization step size however. The bit-stream size is directly proportional to the number of MP iterations, since all iterations have the same size in a uniform quantization scheme and random signal and dictionary. With in-loop quantization, MP adapts to the quantized coefficient, and the residue corresponds to the stream used for reconstruction. It therefore generally performs better than the *a posteriori* quantization, even if the MP decay can be slightly slower. The difference between both quantization schemes is however very small for large coefficients (where the relative quantization error is kept small). For very small coefficients (i.e., high stream size), the quantization step size is too large to capture the coefficient value. The *a posteriori* quantization error can thus even increase with the number of iterations (i.e., very small coefficients), as in the case of a mid-step uniform quantizer used in the Figure 7.2. However, an in-loop quantization scheme stays less flexible than the *a posteriori* quantization since the stream is generally targeted for a given rate.

The aim of quantization is to offer the best possible reconstruction quality for a given bit budget. The *a posteriori* quantization based on the upper-bound of the coefficients proposes a rate-distortion optimal solution taking benefit from the exponential decay of the residual energy. This interesting property directly drives the quantization of the coefficients in two ways. Intuitively, the quantization error on a Matching Pursuit element depends first on the iteration number. The highest iteration elements indeed bring less energy than the first elements. Their quantization upper-bound can thus be reduced compared to the upper-bound of the high energy coefficients. Second, the number of Matching Pursuit elements can also be adapted to the available bandwidth, by dropping the lowest energy elements.

Let c_{γ_n} represent the scalar product $\langle \mathbf{g}_{\gamma_n}, \mathbf{R}^n \mathbf{s} \rangle$. From Eq. (7.3), its norm is upper-bounded by

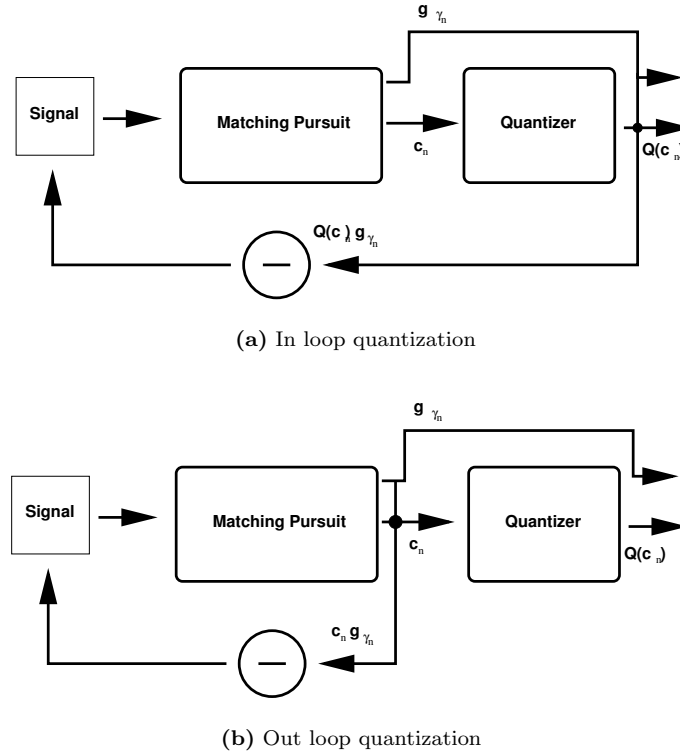


Figure 7.1: In loop and out loop image quantization schemes. The in loop MP quantization scheme subtracts to the original signal the quantized projection, while the out-loop uses the non-quantized values for image subtraction.

an exponential function, which can be written as

$$|c_{\gamma_n}| \leq (1 - \alpha^2 \beta^2)^{\frac{n}{2}} \|s\|. \quad (7.7)$$

The upper-bound depends on both the energy of the input function and the construction of the dictionary. Since the coefficient can obviously not bring more energy than the residual function, the norm of the coefficient is strongly related to the residual energy decay curve. Figure 7.3 represents the energy of the Matching Pursuit coefficients for the expansion of random signals over a 128-random vector dictionary ($\alpha = 1$). It clearly illustrates that the coefficients norm can be upper-bounded by an exponentially decaying curve. This qualitative result does not depend either on the signal or on the dictionary. However, the decay rate directly depends on the redundancy factor β as presented in previously in this chapter.

The exponential upper-bound on the coefficients can be used to design an efficient quantization scheme [75]. In the following discussion, an Oracle computation of the structural redundancy factor of the Matching Pursuit dictionary is assumed, and this redundancy factor is used to derive an efficient quantizer. Later on, an adaptive algorithm that avoids the computation of such a factor is presented. The starting point for the design of an efficient quantizer is from observation that there is clearly no need to quantize all coefficients on the same range, since their values decrease exponentially. In other words, the quantization applied to the first coefficients is certainly not efficient on the last ones, as their range is (exponentially) smaller. Bits can thus be saved by simply limiting the quantization region between 0 and the exponential decay curve given by the parameters

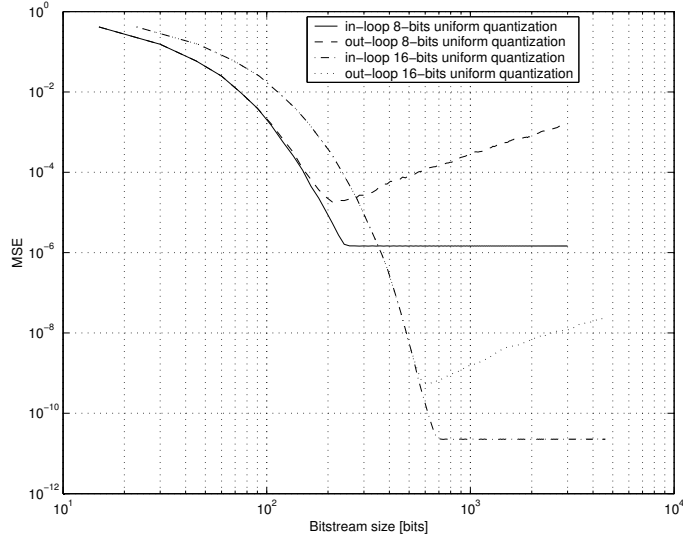


Figure 7.2: Comparison of *a priori* and *a posteriori* uniform coefficient quantization of the MP expansion of 10-sample random real signals over a dictionary of 128 random vectors. The distortion have been averaged over 10 independent encodings. To draw this graph, the quantization step size has been kept constant, and both the *a priori* and *a posteriori* quantization schemes have been applied the same quantizer. The increase of distortion at the higher rates comes from the fact that, for high iteration number, the coefficients become smaller than one half of the quantization step, and thus the distortion due to quantization is higher than the energy that these basis functions subtract.

(i.e., β and $\|\mathbf{s}\|$), after possible coefficients sorting. Indeed, even if the coefficient energy is upper-bounded by an exponentially decaying curve, the coefficient magnitude is not necessarily strictly decreasing due to the MP algorithm and its implementation. The quantized coefficients are then sent to the entropy coder together with an additional bit of sign.

The *a posteriori* MP coefficient quantization allows to compute the optimal number of coefficients, as well as the optimal number of bits per coefficient in exponentially upper-bounded quantization. The distortion at decoding can be bounded by the sum of the quantization error and the approximation error due to the limit on the number of Matching Pursuit iterations. Let $\xi_n = c_{\gamma_n} - \tilde{c}_{\gamma_n}$ denote the error on the coefficient. The total distortion D can thus be written as:

$$D \leq \sum_{n=0}^{N-1} |\xi_n|^2 + \|\mathbf{R}^N \mathbf{s}\|^2 \leq \sum_{n=0}^{N-1} |\xi_n|^2 + (1 - \alpha^2 \beta^2)^N \|\mathbf{s}\|^2, \quad (7.8)$$

where the energy of the residue at iteration N is bounded thanks to Eq. (7.3).

Assume now that the distribution of the coefficients norm is uniform between 0 and the exponential upper-bound given by Eq. (7.7). This is clearly an oversimplified hypothesis but, as can be seen on Figure 7.4, the exact distribution depends on the iteration number and its properties dramatically change for different coefficients: the first coefficients are quite narrowly distributed close to the upper-bound, while other tend to be more and more compactly distributed near small values. This behavior is mainly due to the very conservative upper-bound defined in Eq. (7.7), which assumes no *a priori* knowledge on the input signal. The simplified uniform distribution model therefore reflects the lack of a real structure of coefficients distribution in this particular context. The uniform quantization is moreover justified in a generic algorithm, where the only available parameters are the signal

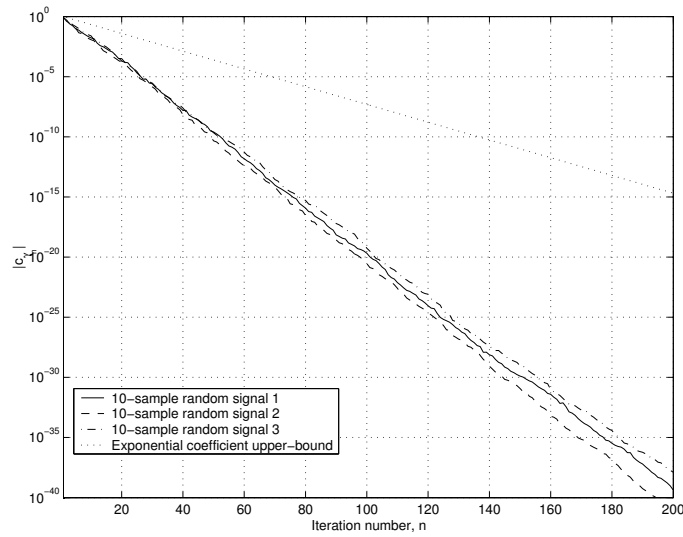


Figure 7.3: Coefficient norm versus the iteration number for the MP expansion of 3 random signals of length 10 over a 128-random vectors dictionary ($\alpha = 1$).

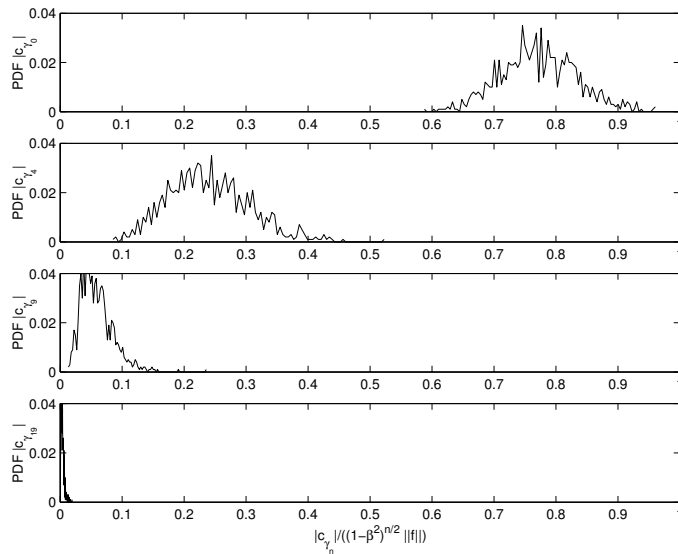


Figure 7.4: Distribution of coefficients $n = \{0, 4, 9, 19\}$ relatively to the exponential upper-bound, for the MP expansion of thousand 10-sample random signals over a random dictionary of 128 atoms ($\alpha = 1$).

energy and the dictionary redundancy, without considering neither the particular relation between the signal and the dictionary or dependency among successive coefficients. For example, one could assume a Gaussian coefficient distribution around an average value decaying faster than the exponential upper-bound. However, the evolution of this average value clearly depend on the efficiency of the dictionary to decompose a particular signal \mathbf{s} . This relation is shown later in this chapter. Finally, it can be observed that successive coefficient norms are likely to be close, and the distance to the upper-bound increases with the coefficients order, since the upper-bound is clearly conservative. The adaptive algorithm proposed in the next section will take benefit of these observations.

Under the previous assumptions, the coefficient c_j is uniformly quantized within the exponentially decaying quantization range $I_j = \nu^j \|\mathbf{s}\|$, where $\nu = (1 - \alpha^2 \beta^2)^{\frac{1}{2}}$. Let n_j be the number of quantization steps within $[0, I_j]$ for the quantization of the j^{th} coefficient. Similarly, let a_j represent the number of bits needed to code the atom index. The atom indices can be entropy coded to improve the compression and a_j represents in this case the length of atom j codeword, without loss of generality. The coding rate for N MP atoms is therefore given by:

$$R = \sum_{j=0}^{N-1} \log_2(n_j) + \sum_{j=0}^{N-1} a_j. \quad (7.9)$$

Also, in the case of fine uniform quantization, the distortion can be written as:

$$D_Q = \sum_{n=0}^{N-1} |\xi_n|^2 = \sum_{j=0}^{N-1} \frac{\nu^{2j} \|\mathbf{s}\|^2}{12 n_j^2}. \quad (7.10)$$

The optimal quantization problem, which minimizes the distortion for a given bit budget R_{budget} can now be formulated as follows :

Problem 7.1 *Given a Matching Pursuit expansion represented by a set of coefficients c_j and their respective atoms \mathbf{g}_{γ_j} , coded on codewords of length a_j , find N , the number of atoms, and $\{n_j\}$, with $0 \leq j \leq N-1$, the number of bits needed to code the respective coefficients, such that the distortion D is minimized, while the coding rate is smaller than the bit budget, i.e., $R \leq R_{\text{budget}}$.*

The Lagrangian multiplier method [58, 141] is well suited for this kind of constrained optimization problems. It defines a cost function $\mathcal{L}(\lambda)$ as the sum of the objective distortion function and the constraint on the rate, weighted by the Lagrangian multiplier λ . This formulation allows to solve the hard constrained problem of finding the optimal set of n_j and N by converting it to a set of unconstrained problems driven by λ . In our case the cost function can be written as:

$$\mathcal{L}(\lambda) = D + \lambda R = \sum_{j=0}^{N-1} \frac{\nu^{2j} \|\mathbf{s}\|^2}{12 n_j^2} + \nu^{2N} \|\mathbf{s}\|^2 + \lambda \left(\sum_{j=0}^{N-1} \log_2(n_j) + \sum_{j=0}^{N-1} a_j \right). \quad (7.11)$$

The optimal quantization is obtained by differentiating $\mathcal{L}(\lambda)$ with respect to both n_j and N . First, solving

$$\frac{\partial \mathcal{L}(\lambda)}{\partial n_j} = 0, \forall j \quad (7.12)$$

for n_j positive and finite, yields

$$n_j = \sqrt{\frac{\|\mathbf{s}\|^2 \nu^{2j} \log 2}{6\lambda}}. \quad (7.13)$$

The solution of Eq. (7.13) is a minimum of the Lagrangian since the second derivative is positive at this point, regardless of the value of j . Hence, the optimal quantization imposes an exponential law on the number of quantization levels:

$$\frac{n_{j+1}}{n_j} = \frac{\nu^{j+1}}{\nu^j} = (1 - \alpha^2 \beta^2)^{\frac{1}{2}}. \quad (7.14)$$

Interestingly, this previous relation leads to an equivalent participation of each iteration to the total distortion. Indeed, the average distortion per coefficient is equal to:

$$\frac{\lambda}{2 \log(2)}, \quad \forall n_j > 1, \quad (7.15)$$

independently of the iteration. Notice however that the Lagrangian formulation provides only an approximation to the optimal quantization. Indeed, in the practical case, n_j can only take integer values, moreover often limited to integer powers of two.

The coding rate can also be limited by transmitting only part of the MP expansion. On the one hand, even if small coefficients can be efficiently approximated by the exponential upper-bound, the quantization scheme may decide not to code them because the cost of the atom index is too expensive. On the other hand, it can be seen from Eq. (7.11) that atoms may be transmitted alone, even if no coefficient is coded ($n_j \leq 1$). Depending on the redundancy of the dictionary, atom indices often carry more information than coefficients. With efficient entropy coding, the atom indices may moreover become very cheap to code. One can therefore imagine a scheme where the coefficients are simply interpolated from the exponential decay curve, especially for high order iterations (i.e., small coefficients). The optimal number of iterations N is thus given by minimizing the Lagrangian cost function of Eq. (7.11) where n_j has been replaced by its optimal value from Eq. (7.13):

$$\mathcal{L}(\lambda) = \sum_{j=0}^{N-1} \mathcal{L}_j(\lambda) + \nu^{2N} \|\mathbf{s}\|^2 \quad (7.16)$$

with

$$\mathcal{L}_j(\lambda) = \begin{cases} \frac{\nu^{2j} \|\mathbf{s}\|^2}{12 n_j^2} + \lambda (\log_2 n_j + a_j) & \text{if } n_j \geq 1 \\ \frac{\nu^{2j} \|\mathbf{s}\|^2}{12} + \lambda a_j & \text{otherwise.} \end{cases} \quad (7.17)$$

Because indices can be transmitted without coefficients, the Lagrangian is defined as a piecewise function. The break-point occurs at $j = N_B = \lfloor \frac{-\log(n_0)}{\log(\nu)} \rfloor$ (i.e., $n_j = 1$) where $n_0 = \sqrt{\frac{\|\mathbf{s}\|^2 \log(2)}{6 \lambda}}$ from Eq. (7.13). Notice that N_B is positive only if $\lambda \leq \frac{\|\mathbf{s}\|^2 \log(2)}{6}$. Otherwise, the weight on the rate in the Lagrangian cost function becomes much more important than the distortion, and the best scheme would be not to transmit any coefficient. Let now assume that $a_j = A$ is constant, where A correspond to the average size of the atom indices, after possible entropy coding. The optimal number of iterations, N is given by the following theorem:

Theorem 7.1 *The Lagrangian cost function of Eq. (7.16) has a unique minimum in terms of the number of MP iterations and this optimal solution is given by:*

$$N = \begin{cases} N_1 & \text{if } A > -\frac{23 \log(\nu) + 1}{2 \log(2)} \\ N_2 & \text{otherwise.} \end{cases} \quad (7.18)$$

where:

$$N_1 = -\frac{1 + 2 A \log(2) - \log(\nu) + \log\left(\frac{\|\mathbf{s}\|^2 \log(2)}{6 \lambda}\right)}{2 \log(\nu)} - \frac{W_{-1}\left(3 \cdot 2^{3-2A} \nu \log(\nu) e^{-1}\right)}{2 \log(\nu)}, \quad (7.19)$$

$$N_2 = \frac{\log\left(-\frac{\lambda A}{2\|\mathbf{s}\|^2 \log(\nu) \left(1 + \frac{1}{12 \nu^2 - 12}\right)}\right)}{2 \log(\nu)} \quad (7.20)$$

are the optimal values of N on both sides of the piecewise function (i.e., for respectively $N \leq N_B$ and $N > N_B$).

The choice between N_1 and N_2 depends only on the relation between A and ν . Indeed, the atom index size and the decay rate of the coefficient norm drive the decision of coding an atom without coefficients. In Eq. (7.19), $W_{-1}(x)$ represents the second branch of the Lambert W function [25].

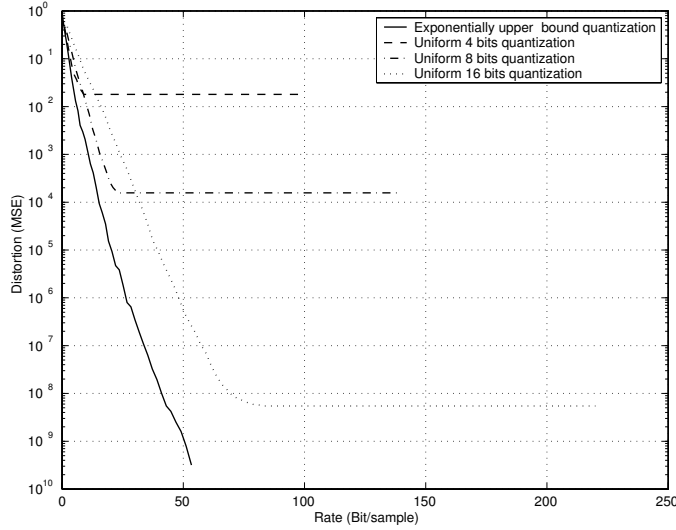


Figure 7.5: Average R-D curve for the MP expansion of ten 10-sample random real signals over a 50-vector random dictionary, in the case of both exponential and uniform quantization.

Finally, the solution of Eq. (7.18) is a minimum of the Lagrangian cost function, since the second derivative is positive at N (see [72, 75] for a detailed proof).

Now that the unconstrained problem of Eq. (7.11) has been solved optimally for an arbitrary λ , the next step is to find the optimal λ_{opt} that guarantees a bit budget R_{budget} . The solutions of the optimization problem form a convex hull of the achievable rate-distortion curve, and the bit budget constraint imposes a λ_{opt} that represents the slope of the rate-distortion function at $R = R_{budget}$. Several bisection methods have been proposed to solve this second problem [151, 166, 171]. Since in our case the bit rate is expressed as a function of the Lagrangian multiplier λ , the constraint on the bit budget directly imposes an approximated value for λ_{opt} from:

$$R_{budget} = \begin{cases} \sum_{j=0}^{N_1-1} \log_2(n_j) + N_1 A, & \text{if } A > -\frac{23 \log(\nu)+1}{2 \log(2)} \\ \sum_{j=0}^{N_B-1} \log_2(n_j) + N_2 A, & \text{otherwise.} \end{cases} \quad (7.21)$$

where N_1 , N_2 and n_j are functions of λ from the above equations. Recall that the approximation is due to the constraint of integer number of bits for each coefficient. The values of λ_{opt} can finally easily be computed through numerical methods.

Figure 7.5 shows the rate-distortion characteristic of the MP expansion of a 10-sample random signal over a 50-vector dictionary. The redundancy-driven quantization scheme is compared to different lower-step uniform quantization schemes, which simply quantize the coefficients over a range defined by the energy of the signal. The proposed exponentially upper-bounded quantization clearly outperforms the uniform quantization, since it adapts to the rate and to the range of the coefficients to provide the best approximation for the available rate. Uniform quantization schemes provide good results for low rate, since the first coefficients are finely quantized. For high rates, however, the coefficients become too small compared to the quantization steps and they are set to zero. The error may even increase in the case of mid-step quantization, since the quantized coefficient becomes larger than the true value computed by the MP (see Figure 7.2). Nevertheless,

for a number of coefficients small compared to the dimensionality of the signal and an appropriate quantization step, this case does not normally appear.

Finally, it can be noted that the exponentially upper-bounded uniform quantization is equivalent to the division of the coefficients by an exponential quantization table factor, multiplied by a quantizer scale factor which is given by the bit budget. Practically speaking, the only parameters to pass to the decoder are ν , $\|\mathbf{s}\|$ and n_0 , or equivalently the quantizer scale factor. The optimal number of iterations thus only depends on the bit budget and the design of the input dictionary. The redundancy-driven quantization becomes particularly interesting for highly redundant dictionary, where the coefficient value decreases very rapidly. In this case, the distribution of bits among coefficients is particularly efficient compared to uniform quantization.

The scheme proposed above heavily relies on the knowledge of the structural redundancy factor exponential decay factor ν . We call this scheme Oracle-based because, in practical situations, it can be too demanding to compute this parameter. Moreover its influence is mixed with the suboptimality factor α brought in by MP implementation choices. This additional factor is moreover very difficult to control. In the next section we develop an adaptive algorithm that uses the previous theory to dynamically estimate the quantization parameters, therefore relaxing assumptions on the *a priori* knowledge of the redundancy factor.

7.3.2 Practical Scheme: Adaptive Quantization

The previous optimal quantization scheme has several limitations, particularly in the practical case of large dictionaries. Moreover, the accuracy of the exponential upper-bound is highly dependent on the Matching Pursuit algorithm implementation. To overcome these limitations, we now propose a suboptimal but very efficient algorithm based on the development of the previous section. The key idea lies in a dynamic computation of the redundancy factor β (i.e., the parameter ν) from the quantized data. Since this information is also available at the decoder, it will be able to perform the inverse quantization without any additional side information. As in the previous section, we only focus on the *magnitude* of coefficients, reporting their sign on an separate bit.

The MP coefficients are first re-ordered, and sorted in the decreasing order of their magnitude. This operation may be necessary since the MP algorithm does not guarantee a strict decay of the coefficient energy, it just guarantees a strict decrease of the upper-bound of the energy. The adaptive quantization scheme then performs as follows. Let $Q[c_k]$, $k = 1, \dots, j - 1$ denote the quantized counterparts of the $j - 1$ first coefficients. Due to the rapid decay of the magnitude, coefficient c_j is very likely to be smaller than $Q[c_{j-1}]$. It can thus be quantized in the range $[0, Q[c_{j-1}]]$. The number of quantization levels at step j is theoretically driven by the redundancy factor as given by Eq. (7.14). The adaptive quantization uses an estimate of the redundancy factor to compute the number of quantization levels as:

$$n_j = \tilde{\nu}_{j-1} n_{j-1}. \quad (7.22)$$

The estimate of the redundancy factor $\tilde{\nu}$ is recursively updated, replacing the quantization range I_j of the optimal quantization scheme with the quantized coefficient $Q[c_j]$ as:

$$\tilde{\nu}_j = \left(\frac{Q[c_j]}{\|\mathbf{s}\|} \right)^{1/j}. \quad (7.23)$$

The adaptive quantization is described by Algorithm 7.1. The quantization is completely determined by the choice of n_0 , the number of bits for the first coefficient, and a positive value of N . The algorithm iteratively quantizes each of the coefficients, and updates, at each step, the estimated value of the redundancy factor. It finally stops after a predefined number of atoms N has been

reached, or when it determines that the number of bits to code a coefficient becomes null (i.e., $n_j = 0$).

Algorithm 7.1: Adaptive Quantization	
Data: $I_0 = \ \mathbf{s}\ $, $n_0 = \sqrt{\frac{\ \mathbf{s}\ ^2 \log 2}{6\lambda}}$, $\tilde{\nu}_0 = 1$	
1	while $j < \text{max iterations } (N)$ do
2	$n_j = \tilde{\nu}_{j-1} n_{j-1}$;
3	$Q[c_j] \leftarrow \text{Quantize } c_j $ on n_j levels with range I_{j-1} ;
4	$I_j \leftarrow Q[c_j]$;
5	$\tilde{\nu}_j \leftarrow \left(\frac{Q[c_j]}{\ \mathbf{s}\ } \right)^{1/j}$ /*update $\tilde{\nu}$ */
6	;
7	$j \leftarrow j + 1$;
8	end

When the bit-stream has to conform to a given bit budget, the parameter n_0 is computed as follows. First, ν is estimated with Eq. (7.23) by training the dictionary on large a set of signals (e.g., images), encoded with the adaptive quantization algorithm. It can be shown empirically that the estimation quite rapidly tends to the asymptotic value of the redundancy factor. As the dictionary is the same for all the images, this parameter is already know at the encoder and the decoder. The estimation of ν is then used to compute λ as a function of the given bit budget with Eq. (7.21), where N is given by Eq. (7.18) (for highly redundant dictionaries, we generally have $N = N_1$). The value of λ determines the number of bits of the first coefficient n_0 , and the number of atoms to be coded, N , using appropriately Eq. (7.19) or Eq. (7.20). The adaptive quantization algorithm is finally completely determined, and generally yields bit rates very close to the bit budget. For high bit-rates, it sometimes under-estimates the quantizer efficiency, due to the fact that the quantization range estimated from the quantized coefficient values is more accurate than the one obtained from the theoretical exponential upper-bound. In this case, the actual bit rate becomes smaller than the bit budget, which can however easily be met by coding additional coefficients.

Notice that, a loose estimation of ν or λ will not impair the efficiency of the quantization algorithm, but rather displace the resulting encoding on the working rate-distortion curve. Finally, several coefficients could be used for the computation of $\tilde{\nu}$ in Algorithm 7.1. Such a modification will improve the accuracy of the estimation of quantization parameters, and avoid potential oscillatory effects.

Experimental Results on random signals

In this section we now compare the adaptive entropy-constrained quantization scheme with an exponential quantization scheme [179] used in Matching Pursuit coding [137]. The exponential quantization is clearly expected to provide better results than uniform quantization [73] due to the particular distribution of the MP coefficients. Both the exponential and the adaptive entropy-constrained quantization schemes, comparable in terms of complexity, are used for *a posteriori* MP coefficients quantization. In the particular case of random signal coding, the distortion is reported to the coding rate estimated as the sum of the quantized coefficients entropy and the index average size. Note that the exponential quantization is similar to the one proposed in [137], where the dead-zone is adapted to the statistics of the coefficients.

Figure 7.6 shows the evolution of the MSE distortion versus the coding rate for both the adaptive entropy-constrained and the exponential quantization. The curve has been averaged on one hundred

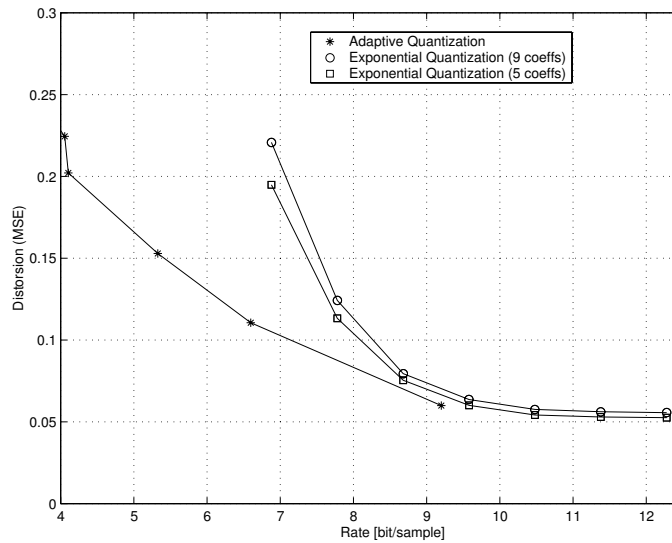


Figure 7.6: Rate-distortion curve for adaptive and exponential quantization the MP expansion of 10-sample random real signals over a 128-atom dictionary.

MP expansions of a random \mathbb{R}^{10} source. The adaptive scheme clearly provides better results since it optimally distributes bits among MP coefficients. It moreover allows to reach much lower coding rates, since the number of coefficients is adaptively chosen according to the bit budget. A more detailed comparison of these two quantization schemes can be found in [75]. The results of applying this quantization scheme to image coding are shown in the next section.

7.3.3 Probability distribution optimized MP coefficient quantization

Until here, it has been seen that taking into account the theoretical MP coefficient energy upper-bound for quantization gives better R-D results than schemes that do not take this into account. This section studies whether these results can be improved by studying not only the theoretical energy upper-bound, but the real probability distribution of the coefficient value with respect to the iteration number.

The empirical data for estimating the probability distribution and the first moment of the MP coefficients with respect to the iteration number is needed to optimize the quantization scheme of the MP coefficients. In order to have a proper estimation of the parameters, 500 MP coefficients from 88 128x128 natural images were obtained. This allowed to estimate the mean evolution of the MP coefficients, and to verify whether this evolution of the mean of the MP coefficients and the evolution of a single image MP coefficient stream with the iteration number were similar. Results were encouraging, because the evolution of a single coefficient and the evolution of the mean showed a clearly linear behavior when using the log-log scale.

Taking this into account, a theoretical expression of the mean value coefficient n when coefficient $(n-1)$ is known can be easily expressed mathematically. As it is linear in the log-log scale, in the linear domain the coefficients evolve with the iteration number as:

$$c_n = an^{-\varpi} + b, \quad (7.24)$$

with ϖ a real positive number. The above expression, when translated to the log-log scale gives a

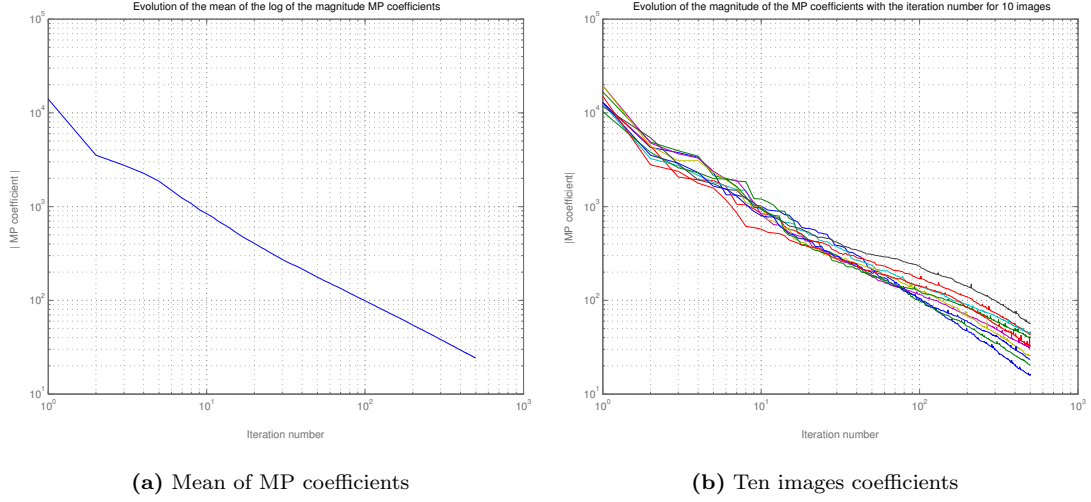


Figure 7.7: (a) Evolution of the mean of the log of the MP coefficients with the log of the iteration number. It can be seen that the behavior is nearly linear. The mean was estimated from 88 different images. (b) Evolution for a single image coefficient stream of the log of its coefficients with the log of the iteration number. It can be seen that for a single image, the behavior of the coefficients does not differ from the evolution of the mean.

linear behavior:

$$z_i = u - \varpi i. \quad (7.25)$$

In the above expression, $i = \ln n$ is the logarithmic iteration number and $z_n = z_{\ln n} = \ln(c_n - b)$ is the logarithmic magnitude of the coefficient, shifted by a parameter b .

With the above expression, it is easy to prove that:

Theorem 7.2 *If z_i is the logarithmic magnitude at logarithmic iteration n and ϖ a valid slope, then the logarithmic magnitude at logarithmic iteration $n + 1$ is:*

$$z_{i+1} = z_i - \varpi, \quad (7.26)$$

with ϖ the characteristic value of process z_n .

The accuracy of the above theorem is experimentally proved, through results shown in Figure 7.7.

In order to have a better understanding about the probability distribution of the evolution of the energy of the projection with the iteration number, it is useful to compute the first and second moment of z_i , in addition of the mean. The first moment is straightforward to compute, due to the linearity of all the operations that are involved. Being $E\{z_i\} = \mu_{z_i}$, $E\{u\} = \mu_u$, and $E\{\varpi\} = \mu_\varpi$, it can be seen that:

$$E\{z_i\} = E\{z_{i-1}\} - b = \mu_u - i\mu_\varpi. \quad (7.27)$$

As can be seen in Figure 7.9, the mean value of ϖ does not depend on the iteration number, thus in the above expression b is independent of the iteration number.

The second moment is computed by:

$$\sigma_{z_i}^2 = E\{(z_i - \bar{z}_i)^2\} = E\{z_i^2\} - \mu_{z_i}^2. \quad (7.28)$$

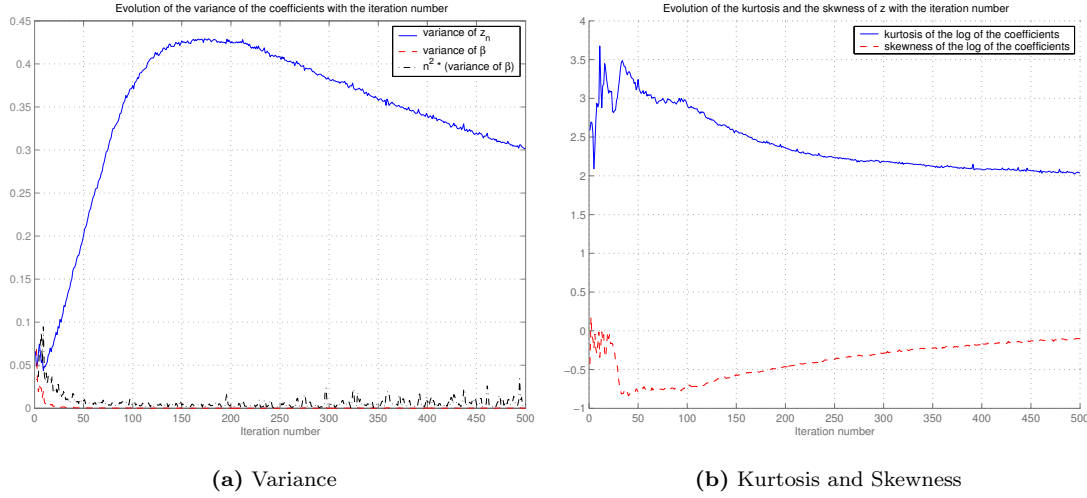


Figure 7.8: (a) Solid line: Evolution of the variance of z_i with the iteration number. Dashed line: Evolution of the variance of ϖ with the iteration number. Dot-dashed line: Evolution of $i^2 \sigma_{\varpi}^2$ with the iteration number. (b) Evolution of the kurtosis (solid line) and the skewness (dashed line) of z .

Substituting z_i by the expression given in Eq. (7.25), one can easily compute that the variance of z_i is given by:

$$\sigma_{z_i}^2 = \sigma_u^2 + 2i(\mu_u \mu_{\varpi} - E\{u\varpi\}) + i^2 \sigma_{\varpi}^2. \quad (7.29)$$

From Figure 7.8, one can see that $i^2 \sigma_{\varpi}^2$ is, after the first iterations, very small and almost constant, and thus does not really affect the value of $\sigma_{z_i}^2$. As μ_u does not depend on the iteration number, because it is the variance of the first MP coefficient, the variance of z_i will evolve with n according to the term $2n(\mu_u \mu_{\varpi} - E\{u\varpi\})$. This means that μ and ϖ are not independent, which is logical, because it is the term that somehow relates the dictionary and the signal to represent. $E\{u\varpi\}$ will be a sort of measure of how the dictionary adapts to the signal to code.

The distribution of the coefficients around the mean is not exactly Gaussian, as can be seen by the estimated Skewness and Kurtosis. For a Gaussian distribution, the Skewness should be 0 and the Kurtosis, 3. This values of Kurtosis and Skewness show that the coefficient distribution around the mean value is slightly asymmetric, skewed to the smaller coefficient values, and slightly flatter than a normal distribution.

Some remarks about the coefficient evolution

Several deductions can be driven from the coefficient distribution and its evolution with the iteration number. The first thing that can be seen is the linear evolution of the mean of the coefficients with the logarithm of the iteration number. This linear evolution has to be taken carefully, though, because it is not valid for infinite coefficients, it is just valid for a limited and small (compared to the dimensionality of the signal to represent) number of coefficients. As the coding scheme presented here works exactly in this range, this observations are appropriate for the data under study.

From this observations, we can deduce:

- The mean of the coefficients is decreasing following an exponential. The quantizer designed with the upper bound of the coefficients does not take this into account, and performs a step-like quantization. Knowing this exponential strict decreasing (not only the upper-bound,

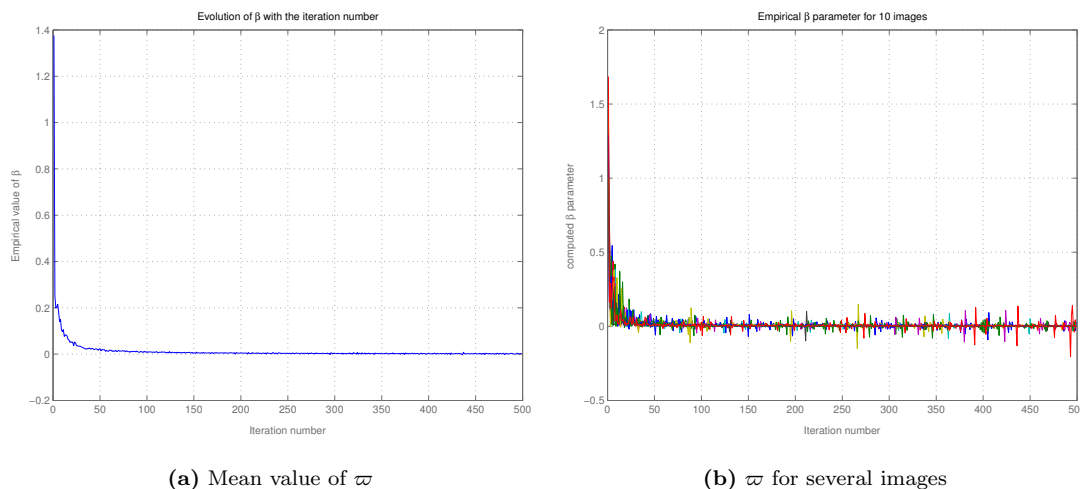


Figure 7.9: (a) Evolution of the empirical mean value of ϖ with the iteration number. (b) Evolution of the value of ϖ for ten image coefficient streams.

but actually the real coefficient behavior is exponentially decreasing), one can, with the same number of bits, reduce the distortion introduced by the quantizer by taking this fact into account.

- The difference of consecutive coefficients has a much smaller variance than the actual coefficient distribution, and thus, less bits are needed to code this difference. It is worth, then, performing a differential coefficient coding, instead coding the value of the coefficient at each iteration.
- The kurtosis and the variance of the coefficients indicate that these do not follow a Gaussian distribution, and thus, standard techniques based on Gaussian distributions cannot be applied to improve the coding of this coefficients.

The study of the distribution of the coefficients indicated that the quantizer based on the theoretical upper-bound of the coefficients is useful as a quantization, but that the value that has to be sent to the entropy coder is not the direct output of this quantizer, but a differential coding (quantized value of the previous coefficient minus quantized value of the present coefficient). In addition, the quantization noise can be reduced by applying the observations of the mean of the coefficients to the quantized data, and make this data closer to the real behavior of the coefficients. This will allow to increase the quantization step while reducing the distortion due to quantization of the coefficients. This allows to design an improved decoding scheme for MP streams is presented, based on the *a posteriori* MP quantization method. This improved quantized and dequantizer scheme is presented in the next section.

7.3.4 Reconstruction of MP Coefficients by Interpolation

By observing the output of the dequantizer it can be noticed that the output values are relatively coarsely quantized for atoms with high iteration numbers (Figure 7.10). It is known that coefficient magnitudes are always decreasing (when eventually the decreasing of the coefficients is not strict, a reordering can be done), and the coefficient decay follows a very well defined pattern. It is thus possible to estimate the original values and obtain a better quality for the reconstructed image.

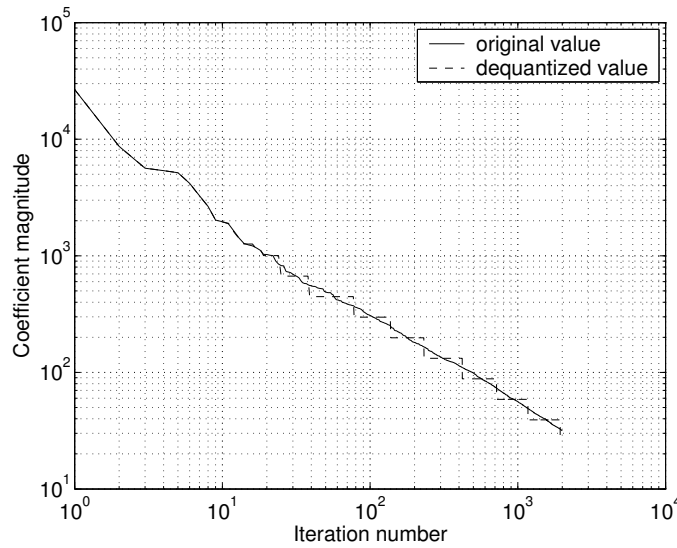


Figure 7.10: Image *Goldhill* (256x256): Behavior of unquantified coefficient magnitudes in log-log scale compared to dequantized values using 1024 initial quantization levels.

Figure 7.10 shows the behavior of the coefficient magnitudes in the log-log scale for one single image. It can be observed that this behavior is approximately linear in log-log scale starting from a certain iteration number, even for a single image case. This linearity has been observed for most of the images studied. Therefore, the logarithmic coefficient magnitudes and the logarithmic iteration number can be approximated by the expression given in Eq. (7.25).

Calling \hat{c}_n the estimated coefficient at iteration n , one can estimate the parameters a , ϖ and b of Eq. (7.24) by minimizing the quadratic error $(c_n - \hat{c}_n)^2$ (with c_n the actual coefficient value and \hat{c}_n the estimated one). Parameter b is introduced to model slight changes in the linearity for different iteration numbers, which is quite usual in a single image coefficient evolution. Figure 7.11 shows the original coefficient value and its approximation, computed with Eq. (7.24). The estimation has been performed from the iteration 120 to 1000. It can be seen that the approximated values obtained with such a method are very close to the real ones, thus, this property can be used to improve the compression rate achieved with the MP coder. The question that arises now is how to parametrize and quantize the parameters ϖ , b and a . Any quantization of these parameters causes a big loss of accuracy, and thus, some method to avoid directly sending them has to be found.

One can naturally think to use the well known *a posteriori* quantization scheme based on the exponential upper bound and improve it with the knowledge of the log-log linearity of the MP coefficients. The quantization scheme based on the exponential coefficient upper-bound, when few quantization steps are used, gives not a smooth output, but a piecewise flat output (see Figure 7.12). Knowing that the evolution of the MP coefficients has to be smooth, one can allow the *a posteriori* quantizer to have a coarse quantization step (and this give a piecewise flat output) and apply a sort of post-processing of the coefficients in order to reduce, at the receiver, the distortion due to quantization.

Observing the flat constant regions, one can see that the most precise information about the coefficient magnitude values can be obtained from edge points (where the dequantized values change): the left edge point, $A = [n_A, c_A]$ and the right one, $B = [n_B, c_B]$. There is just a region that cannot use these edge points: the last region where the stream is cut, for which no precise value of B can be found.

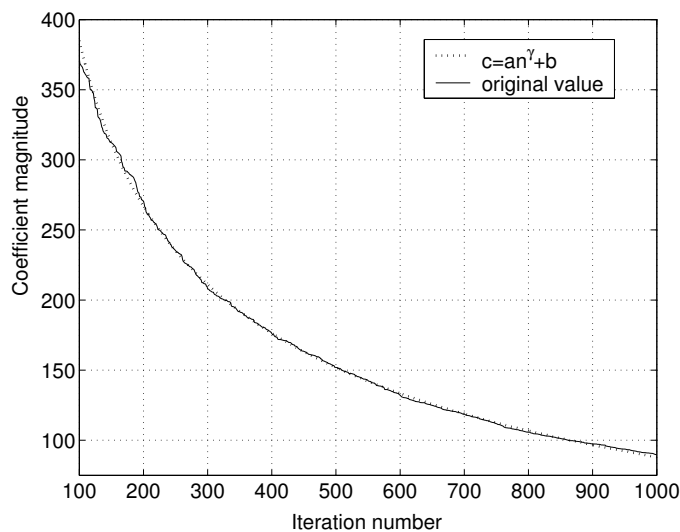


Figure 7.11: Image *Barbara* (256x256): Approximation of coefficient magnitudes with Eq. (7.24); parameters a , b and ϖ are calculated by using a linear regression on original coefficient values for iteration numbers between 120 and 1000.

To compute the coordinates of the edge points, the decoder performs the dequantization and notes two iteration numbers n_1 and n_2 , where the dequantized value c_{deq} changes. The points A and B are then calculated as:

$$n_A = n_1 - 1/2, \quad c_A = \frac{c_{deq}(n_1 - 1) + c_{deq}(n_1)}{2} \quad (7.30)$$

$$n_B = n_2 - 1/2, \quad c_B = \frac{c_{deq}(n_2 - 1) + c_{deq}(n_2)}{2} \quad (7.31)$$

These values represent a very good approximation to the actual values of the coefficient magnitudes (see Figure 7.12; the edge points are represented by asterisks). The decoder knows those edge points and can estimate the values for other iterations by using an interpolation function, better than a constant function. The coefficients that lay in the middle of these two points A and B are then interpolated with Eq. (7.24), substituting with the interpolation function the flat region. This interpolation is performed by fixing one of the three degrees of freedom of the linear function, so that the other two can be estimated using the flat region edge points c_A and c_B . The parameter a and b of Eq. (7.24) are then estimated as:

$$a = \frac{c_B - c_A}{\frac{1}{n_B} - \frac{1}{n_A}}, \quad (7.32)$$

$$b = c_A - \frac{a}{n_A}, \quad (7.33)$$

while the parameter ϖ is fixed to a mean value that has been estimated from a large set of natural images.

In order to detect small variations of the slope of the line, for each constant region the calculation of parameters is independent from neighboring regions, except for the last segment (where the stream is cut), which requires special attention. As the decoder knows only point A , but not point B , the interpolation will be done using the parameters computed from the previous segment,

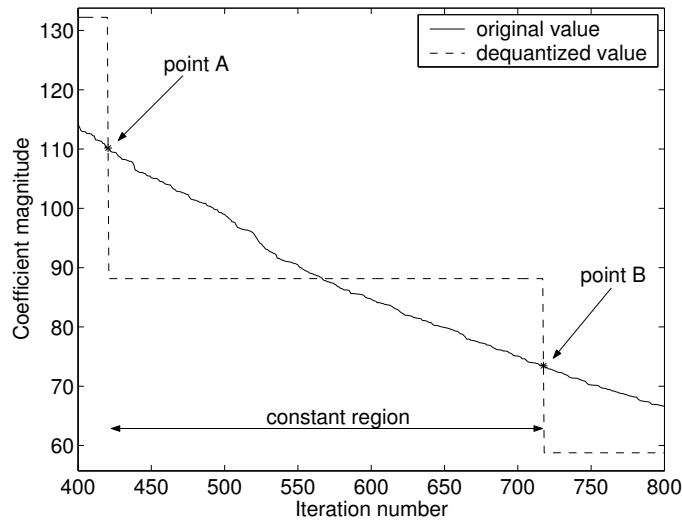


Figure 7.12: Selection of end-points *A* and *B* for interpolating from dequantized values in a given constant region; end points are denoted by asterisks.

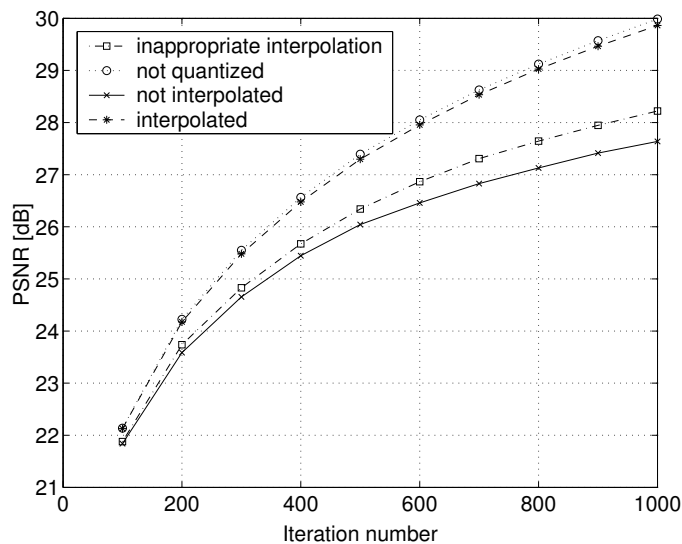


Figure 7.13: Image *Cameraman* (256x256): interpolation with the rational function and 128 initial levels, compared to non-quantized values. Inappropriate interpolation (done before the number of quantization levels becomes low enough) is also shown.

Dequantization results

An example of interpolation using the rational function is given in Figure 7.13, which shows the PSNR as function of the iteration number for the image *Cameraman*. The number of the quantization levels for the first atom is 128, which is *very* small for 1000 iterations. Some remarks need to be given before keeping on the explanation of this scheme. The interpolation explained here performs well only for high iteration number (lower coefficient values). This comes from the fact that coefficients at higher iteration number present a more linear log-log behavior. It is thus important to know which cases should be interpolated, otherwise, this scheme would worsen the quality of the

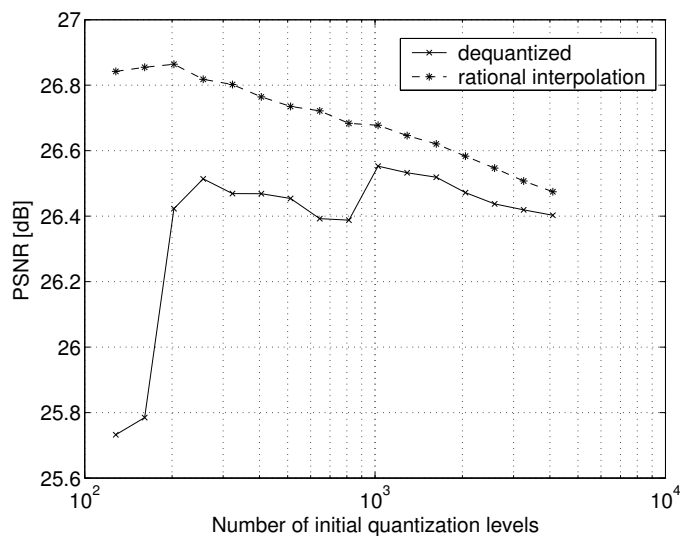


Figure 7.14: Image *Cameraman* (256x256): Optimal number of levels for the first atom, for the given bit rate of 0.2bpp. The number of levels for the first atom ranges between 128 and 4096 with a resolution of 3 values per octave.

reconstructed image instead of improving it. A good method to make sure that the flat zone has to be interpolated is to look at its length: if the flat zone is “long enough”, the coefficients have to be interpolated, otherwise it is better to leave them unchanged. Experimentally, it has been found that the minimum length of the flat zone to need interpolation is approximately five coefficients.

The interpolation scheme does not perform equally well for all images and all numbers of quantization levels. The image *Barbara* with 128 initial levels is such an example. The stream is cut after 1000 atoms. The obtained PSNR for rational interpolation using 128 initial levels is better than using 64 initial levels for most of the iteration numbers, but it deteriorates in the end. The reason for this deterioration is the interpolation for the last segment. For the case of 128 initial levels the final segment starts at iteration number 405 and continues till the end of the stream. The interpolation is based on parameters calculated for the previous segment. Since this last segment is very long, the proposed method does not perform very well. For the case of 64 levels, the last segment starts at iteration number 827 and is much shorter. This last constant segment can be held relatively short by using sufficiently high number of initial quantization levels.

7.4 Matching Pursuit Image Coder structure

7.4.1 Overview

The compression scheme proposed in the scope of this thesis uses the Matching Pursuit decomposition. After a quantization of the coefficients, it sends the atom parameters to an entropy coder, which gives the final bit-stream. A schematic representation of such a coder can be seen in Figure 7.15.

In order to code an image, the coder first recursively decomposes it by Matching Pursuit, in a series of atoms chosen from a redundant dictionary, with their respective coefficients. The coefficients are then quantized by means of an exponentially bounded uniform quantization method adapted to progressive Matching Pursuit stream characteristics, following the quantization scheme introduced in the previous section. Coefficients and atom indexes are finally entropy coded with a context

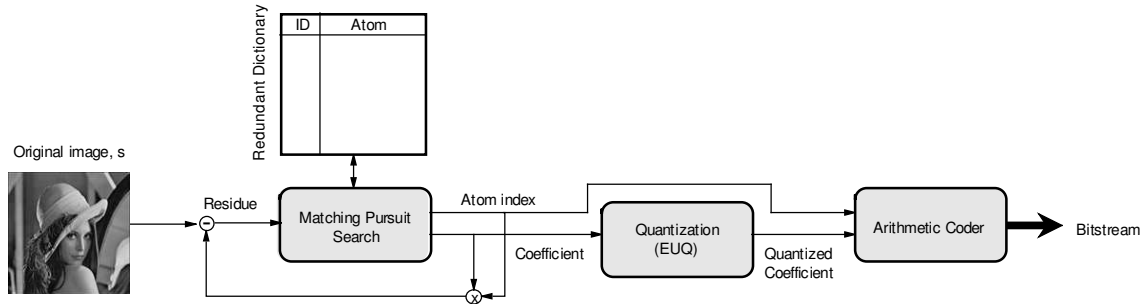


Figure 7.15: Block diagram of the Matching Pursuit image coder.

adaptive arithmetic coder. Quantization and coding are performed *a posteriori*, after the Matching Pursuit expansion. This choice is justified by the fact that the Matching Pursuit algorithm clearly represents the most computationally expensive part of the encoder. If the quantization is performed *a priori* (i.e., in the MP loop), the generated expansion is specifically destined to a target streaming rate. *A posteriori* quantization and coding allow for one single expansion to be encoded at different target rates. The image is finally reconstructed by performing the reverse operations at the decoder.

One of the well-known drawbacks of Matching Pursuit is the computational complexity of the search algorithm, which has to be repeated at each iteration. As it has been seen in the previous chapter, the problem can be alleviated by sup-optimal implementations, based for example on Genetic Algorithms [66], and by heuristics linked to the dictionary. Such methods greatly speed up the search, but often sacrifice in the quality of the approximation. They sometimes get trapped in local minima, and choose atoms which do not always maximize the projection coefficient $|\langle \mathbf{g}_\gamma, \mathbf{s} \rangle|$. Other solutions can be found in efficient implementations of the Matching Pursuit algorithm, taking benefit from the structure of the signal and the dictionary, that might for example be decomposed in incoherent blocks. The actual implementation of the proposed encoder performs a full search over the complete dictionary, and computes all the projections in the Fourier domain [62], as explained in Chapter 6. This tremendously reduces the number of computations in the particular case of our dictionary built on anisotropic refinement of rotated atoms, as described here-below. The number of multiplications in this case only depends on the number of scales and rotations.

7.4.2 First coding scheme

Rate Control

When the bit-stream has to conform to a given bit budget, the quantization scheme parameters n_0 and N are computed as follows. First, ν is estimated with Eq. (7.23) by training the dictionary on a large set of signals (e.g., images), encoded with the adaptive quantization algorithm. It can be shown empirically that the estimation quite rapidly tends to the asymptotic value of the redundancy factor. The estimation of ν is then used to compute λ as a function of the given bit budget R_b which has to satisfy:

$$R_b = \sum_{j=0}^{N-1} \log_2 n_j + \sum_{j=0}^{N-1} a_j = \sum_{j=0}^{N-1} \log_2 \nu^j + N \log_2 n_0 + N A, \quad (7.34)$$

where a_j represents the number of bits necessary to code the parameters of atom c_j (i.e., positions, scales and rotation indexes), and $A = E[a_j]$ represents the average index size. From Eq. (7.13), the value of λ indeed determines the number of bits of the first coefficient n_0 , and the number of atoms

to be coded, N : under the condition that atoms whose coefficients are not quantized (i.e., $n_j < 2$) are not coded, $\nu^{N-1} n_0$ has to be at least equal to 2.

The rate control completely determines the adaptive quantization parameters, and generally yields bit rates very close to the bit budget. For high bit-rates, it sometimes under-estimates the quantizer efficiency, due to the fact that the quantization range estimated from the quantized coefficient values is more accurate than the one obtained from the theoretical exponential upper-bound. In this case, the actual bit rate becomes smaller than the bit budget, which can however easily be met by coding additional coefficients.

Notice that, a loose estimation of ν or λ does not significantly affect the efficiency of the quantization algorithm, but rather displaces the resulting encoding on the working rate-distortion curve. Finally, several coefficients could be used for the computation of $\tilde{\nu}_j$ in Eq. (7.23). Such a modification will improve the accuracy of the estimation of quantization parameters, and avoid potential oscillatory effects.

Coding

The image coder proposed in this paper implements an arithmetic adaptive entropy coder. The core of this coder is based on [210], with the probability update method from [54]. As the distribution of the atom parameters (e.g., positions or scales) is dependent on the image to be coded, the entropy coder initializes the symbol probabilities to a uniform distribution. The atom parameters (projection coefficients, positions, scales, etc.) are coded using independent context. In order to keep the intrinsic scalability features of the bit-stream, the parameters are sent in their natural order, along the decreasing order of the coefficient magnitudes. The bit-stream is simply constructed by concatenating together coded coefficients and atom parameters. This method results in a progressive stream, which can be cut at any point to generate rate scalable streams (see Section 7.6). Scalability is the main motivation for choosing this kind of arithmetic coder, rather than more efficient coders that previously estimate the parameters distribution in order to optimally distribute the bits. Grouping atoms according to their position parameters, for example, would also increase the compression ratio. However, the stream would not be progressive, since the atoms would not anymore be ordered according to their relative importance.

7.4.3 Enhanced coding and decoding scheme

The enhanced coding scheme has then the following form of sending the parameters: For the first iteration, the algorithm sends the real projection value, that will be used as upper-bound for the quantizer, together with the image size. After sending this information, the next thing to send to the receiver is the kind of atom that has been chosen by the algorithm in the first iteration. This information takes only one bit (if the first bit is a one, the atom is a Gaussian, otherwise, it is an AR). The positions of the atoms are sent in raw format (p_x, p_y) , taking as much bits as needed to cover the image size. Then, if the first atom is a Gaussian, the algorithm sends its dilation parameter, and if it is an AR, it sends three integers: $[s_x, s_y - s_x, \theta]$, being s_x the x data needed to compute the x axis scaling factor (that takes the form $2^{\frac{s_x}{N}}$). Instead of sending the raw format of s_y , as it is, by definition, always at least as high as s_x , the difference between the first and the second is sent. Finally, the angle parameter is also sent in raw format.

For the other iterations, the only parameter that is sent in a different way is the value of the projection. The quantization algorithm defined in Algorithm 7.1 is used, but instead of sending to the entropy coder the output of this algorithm, the value that is sent is $c_{n-1} - c_n$, n being the coefficient number that is sent to the entropy coder (together with a sign bit for the n coefficient). This allows

to have more data that is kept constant (as it has been seen in Figure 7.8, the mean of the coefficients has a much lower variance, and thus, needs of less bits to be sent). In addition, the number of initial quantization levels is not computed with the rate-control that comes from the theoretical upper-bound, but from the empirical rate-control coming from the modified dequantization scheme. This strategy of sending the difference of the coefficients instead of the coefficients themselves has as a consequence that the coefficient data vector that is sent to the entropy coder has a lot of repeated values. The adaptive entropy coder can, after some iterations of learning process, use a low number of bits for this repeated values. This improved *a posteriori* quantization gives, for a given Rate-budget, a smaller distortion than the previous scheme, as can be seen in Figure 7.23. The possibility of truncating the bit-stream is still kept, with this new coding and decoding scheme, and thus, the full rate-adaptivity is kept, and even improved, as the number of quantization levels needed to achieve smaller distortion is more constant with this adaptive scheme than with the scheme based uniquely on the MP upper-bound.

7.5 Coding Performance

All the results of this section have been obtained with the coding scheme that does not use the enhanced quantization. At the end, some comparisons of the non-enhanced and the enhanced coding scheme are given.

7.5.1 Benefits of anisotropy

Since Matching Pursuit has already been presented as a valid alternative to wavelets or DCT in low bit rate coding schemes [135], it becomes important to show the real benefits of using an anisotropically refined dictionary to represent natural images. Chapter 6 already introduced the benefits of using anisotropy for achieving sparsity, comparing results obtained with four different dictionaries, but the improvements in sparsity do not necessarily mean improvements in the Rate-Distortion curve. It is interesting now to analyze the penalty of anisotropy on the coding rate. In our coder, the addition of anisotropy induces the cost of coding an additional scaling parameter for each atom. To highlight the coding penalty due to anisotropic refinement, the image has also been coded with the same dictionary, built on isotropic atoms, all other parameters staying identical to the proposed scheme. Figure 7.16 illustrates the quality of the MP encoding of *Lenna*, as a function of the coding rate, with both dictionaries. To perform the comparison, the isotropic and the anisotropic dictionaries are generated with the same generating function and with the same discretization of the parameters (3 scales per octave and an angle resolution of 10 degrees). The anisotropy however implies the coding of one additional scale parameter. It is shown that the dictionary based on anisotropic refinement provides superior coding performance, even with longer atom indexes. The penalty due to the coding cost of one additional scale parameter, is largely compensated by a better approximation rate. Anisotropic refinement is thus clearly an advantage in MP image coding.

7.5.2 Rate-distortion performance

Now that the benefit of anisotropic refinement has been highlighted, the compression performance of the proposed image coder is compared to the state-of-the-art JPEG2000 standard*, and with the SPIHT encoder [158]. Figure 7.17 shows the rate-distortion performance of the three coders for common test images, at low to medium bit rates. It can be seen that Matching Pursuit provides

*All results have been generated with the Java implementation available at <http://jj2000.epfl.ch/>, with default settings (Daubechies 9/7 wavelets and 5 decomposition levels).

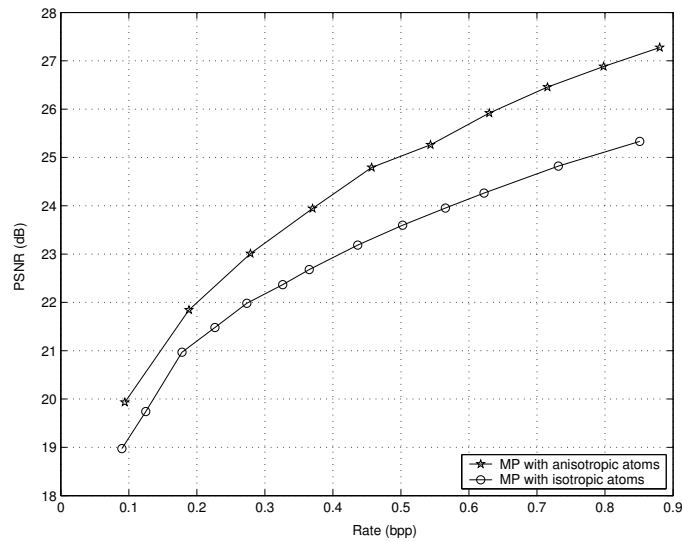


Figure 7.16: Comparison of the rate distortion of anisotropic refinement in the dictionary and its isotropic version. The basis functions used are the same for isotropic and anisotropic functions, with the same parameter discretization, showing the Rate-Distortion improvement of adding anisotropy to the dictionary.

better PSNR rating than JPEG2000 at low coding rates. However, the gap between both coding schemes rapidly decreases when the bit rate increases, as expected. Matching Pursuit and overcomplete expansions are especially efficient for low bit rate coding. They very rapidly capture the most important components of the image, but Matching Pursuit then suffers from its greedy characteristic when the rate increases. Also, the dictionaries have not been designed for texture coding, but rather to code geometry. It has to be noted as well that the bit-stream header penalizes JPEG2000 compared to Matching Pursuit, where the syntactic information is truly minimal (at most a few bits). This penalty becomes particularly important at very low bit rate.

The performance of the proposed coder is also compared to SPIHT encoder, which introduces a minimal syntactic overhead. SPIHT almost always outperforms the proposed coder on the complete range of coding rate, and tends to perform similarly to JPEG2000 for high rates. However, the stream generated by the SPIHT encoder is in general not easily spatially scalable, while MP and JPEG2000 offer increased flexibility for stream adaptation.

Interestingly, all coders give equivalently poor results for highly textured images like *Barbara*. None of them is really optimized for texture coding, and the dictionary used in the proposed MP encoder is primarily designed to capture geometry in images. Finally, the proposed encoder performs reasonably well in terms of rate-distortion performance, especially at low rates. When the rate increases, the saturation of the quality can be explained by the limitations of redundant transforms for high rate coding, and by the dictionary design choices. Hybrid coding schemes could provide helpful solutions at high rate.

7.5.3 Visual comparisons

This section provides visual comparisons of images encoded with the proposed MP encoder, JPEG2000, and SPIHT. Even if the three coding schemes perform similarly in terms of PSNR, as shown in the previous section, the coding artifacts are in general visually less annoying in the MP coding algo-

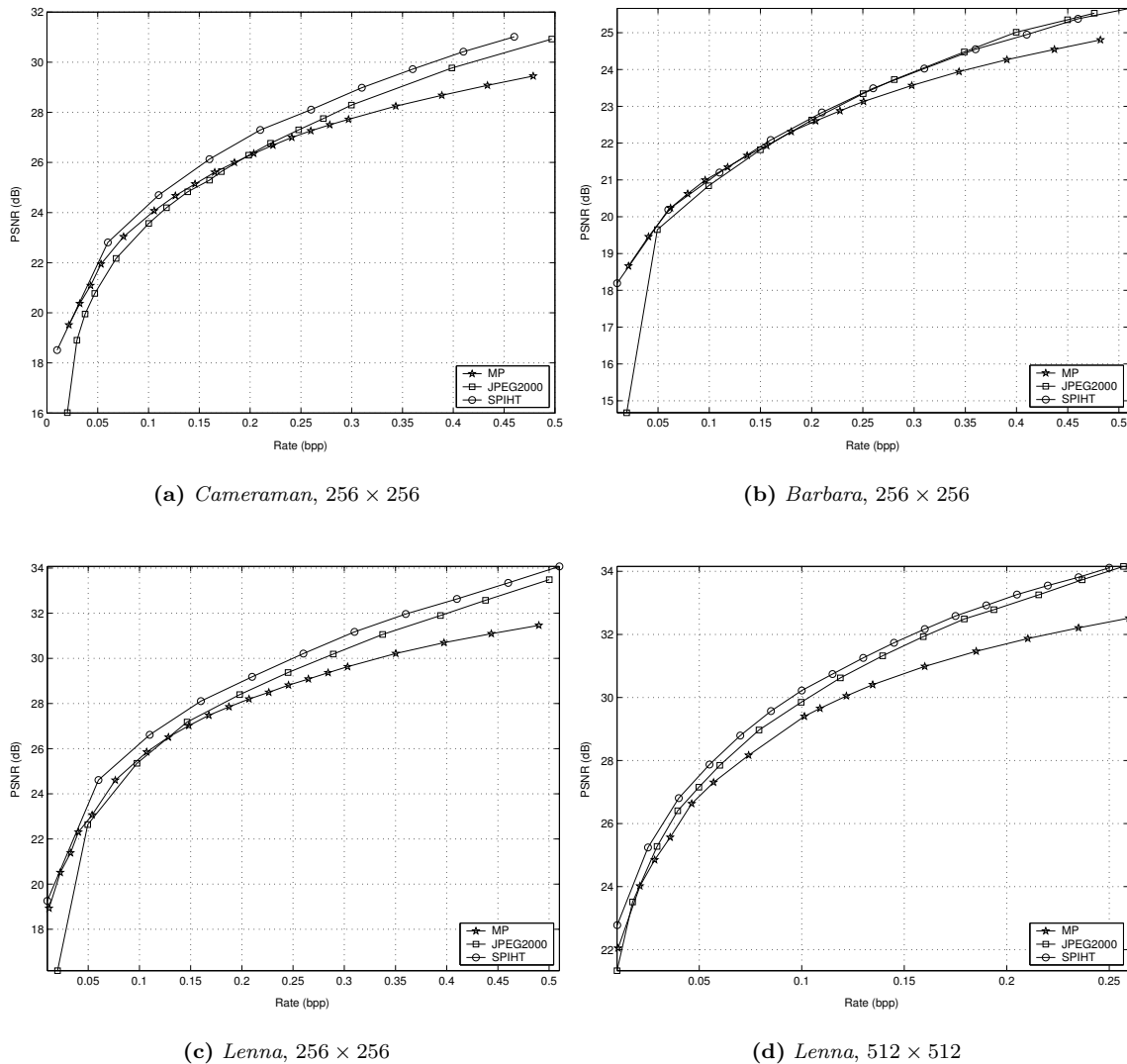


Figure 7.17: Distortion-rate performance for JPEG2000, SPIHT and the proposed MP coder, for several test images.

rithm. Figure 7.18 compares the encoding results for the *Lenna* image, at low bit rate. The decoded images are quite similar in terms of PSNR, but visual comparisons generally seem to favor the MP encoder. The artifacts introduced by MP (basically a simplification of the images) are indeed less annoying than the ringing introduced by the wavelets in JPEG2000 and in SPIHT.

Figure 7.19 presents a similar comparison between MP and JPEG2000, but for a larger image. In this case, the PSNR rating is clearly in favor of JPEG2000. However, the image encoded with MP is visually more pleasant than the JPEG2000 version. The detailed view of the hat, as illustrated in Figure 7.20, clearly confirms this impression. The JPEG2000 encoder introduces quite a lot of ringing, while the MP encoder concentrates its effort on providing a good approximation of the geometrical patterns. JPEG2000 has difficulties to approximate the 2-D oriented contours, which are the predominant components of natural images. Figure 7.21 further confirms the different behaviors of MP and JPEG2000. It can be seen that MP concentrates on edges, and thus rapidly provides



(a) Original *Lenna* 256×256 image.



(b) *Lenna* 256×256 MP image, PSNR=30.3626 dB.



(c) *Lenna* 256×256 JPEG2000 image, PSNR=30.7940 dB.



(d) *Lenna* 256×256 SPIHT image, PSNR=31.3471 dB.

Figure 7.18: *Lenna* 256×256 encoded at 0.35 bpp with MP, JPEG2000 and SPIHT.

a 'sketchy' representation of the image. At the same rate, JPEG2000 already spends some bits in coding textures, while ringing artifacts are still visible.

Finally, Figure 7.22 proposes a comparison of the encoding of a highly textured image (*Barbara*), at a quite low bit rate. As already discussed in the previous section, it can be seen that the three encoders perform quite poorly in this scenario. The type of artifacts introduced by the encoders is however very different. In the wavelet schemes, the low resolution sub-bands provide a low resolution version of the image, but a lot of disturbing noise is present in textured regions. MP first concentrates on well-defined contours, but might fail in coding important semantic parts, like the eyes. Due to the flexibility of MP, one might however propose a modified coding strategy that ensures with priority a low resolution coding of all regions of interest, even if they do not necessarily represent the most energetic features. This modified coding strategy can be implemented through



(a) MP, 31.0610dB



(b) JPEG2000, 31.9285 dB

Figure 7.19: *Lenna* (512 x 512) encoded with MP and JPEG2000 at 0.16 bpp.

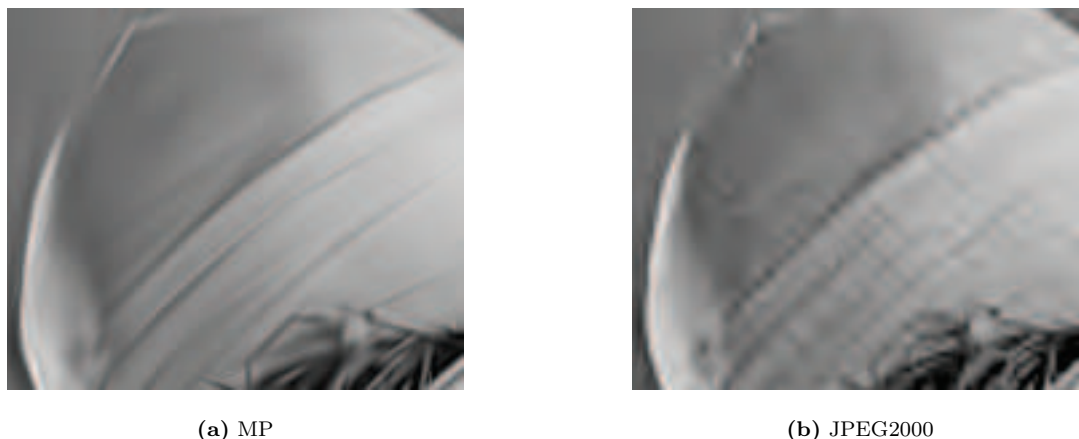


Figure 7.20: Detail view of *Lenna* (512×512) encoded at 0.16 bpp.

the weighted MP scheme presented in Chapter 6, by giving weight 1 to visually important areas (such as faces) and lower weights everywhere else.

7.5.4 Discussion

As shown by the results presented in this section, expansions over dictionaries built on non-separable functions can bring new breakthroughs in image compression. Anisotropic refinement and orientation allow to efficiently capture two-dimensional objects in images, and provide very good performances at low bit rate. In addition to rapidly capturing the most important features of the images, multidimensional representations generate less annoying artifacts than wavelet decompositions, which introduce some ringing at low rate.

Matching Pursuit is however just one method of signal approximation, which shows some limitations at high rate due to its greedy nature. It however allows for quite good signal representations, and presents the advantage of requiring a very low complexity decoding structure. The encoding however still requires quite a lot of computations, and is definitely much more complex than a representation over an orthogonal wavelet basis. A lot of progress have already been made in decreasing the computational complexity of the Matching Pursuit implementation. Schemes based on the division of the dictionary in incoherent blocks for example, already permit to encode images in a few seconds. MP also has the advantage of generating inherently progressive streams, which is a very interesting property for visual communication applications, as discussed in the next Section.

The coding scheme used in this paper can also surely be improved, by taking into account the proper characteristics of the image expansion. The implementation proposed in this paper has been specifically driven by the objective of building highly adaptive bit-streams, but the coding efficiency can certainly be improved when this constraint is relaxed. The PSNR ratings presented in this section are generally better for JPEG2000 than for MP. At the same time, the approximation rate in terms of number of coefficients in the expansion can be shown to be largely in favor of the Matching Pursuit scheme [195].

R-D performance of the enhanced coding scheme

In the enhanced coding scheme, due to the reduction of distortion at the decoder side, the initial number of levels of the *a posteriori* coefficient quantization can be decreased, and more coefficients



(a) MP, 27.49 dB



(b) JPEG2000, 28.18 dB

Figure 7.21: *Goldhill* 256×256 encoded at 0.23 bpp.



(a) Original



(b) MP, 21.35 dB



(c) JPEG2000, 21.23 dB



(d) SPIHT, 21.29 dB

Figure 7.22: *Barbara* (256 x 256) encoded at 0.12 bpp.

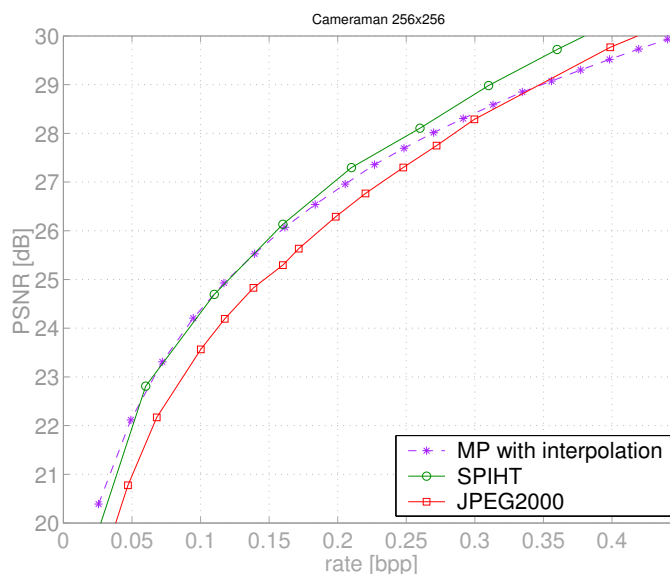


Figure 7.23: Image Cameraman (256x256): The RD performance obtained with the optimal initial number of levels from the Figure 7.14 (203 initial quantization levels with interpolation).

can be sent. Thus, the rate-control scheme presented in Section 7.3.2 can no longer be used. For the modified quantization scheme that takes into account the mean evolution of the coefficients, the optimal number of levels for a given rate cannot be any more theoretically computed, but needs to be estimated. To do this, once the rate has been fixed, a regressive algorithm is used, that performs several quantizations and takes the optimal. This modified quantization scheme does not give a convex R-D behavior with an increasing number of quantization levels, as can be seen in Figure 7.14, and thus, the empirical method is the only way to find the optimal. Nevertheless, as the reconstruction of the MP image is not a computationally hard problem, it is not a problem to have to iterate this procedure. Figure 7.23 shows the improvement that this adaptive quantization and dequantization scheme gives. It can be seen that now the R-D results are better than JPEG2000, and have the same PSNR quality than SPIHT.

7.6 High adaptivity

7.6.1 Importance of adaptivity

As outlined in the previous section, one of the main advantages of the MP coder is to provide highly flexible streams at no additional cost. This is very interesting in nowadays visual applications involving transmission and storage, like database browsing or pervasive image and video communications. We call adaptivity the possibility for partial decoding of a stream, to fulfill decoding constraints given in terms of rate, spatial resolution or complexity. The challenge in scalable coding is to build a stream decodable at different resolutions without any significant loss in quality by comparison to non-adaptive streams. In other words, adaptive coding is efficient if the stream does not contain data redundant to any of the target resolutions.

In image coding, adaptivity generally comprises rate (or SNR-) adaptivity and spatial adaptivity. On the one hand, the most efficient rate adaptivity is attained with progressive or embedded bit-streams, which ensure that the most important part of the information is available, independently

of the number of bits used by the decoder [159, 181]. In order to enable easy rate adaption, the most important components of the signals should be placed near the beginning of the stream. The encoding format has also to guarantee that the bit-stream can be decoded, even when truncated. On the other hand, efficient adaptive coding schemes, like JPEG2000 or the coder proposed in [212] are generally based on sub-band decompositions, which provide intrinsic multiresolution representations. However, spatial adaptivity is generally limited to octave-based representations, and different resolutions can only be obtained after non-trivial transcoding operations.

Multidimensional and geometry-based coding methods can advantageously provide high flexibility in the stream representation and manipulation. In this section, we will emphasize the intrinsic spatial and rate adaptivity of the bit-streams created with our MP image coder. First, due to the geometrical structure of the proposed dictionary, the stream can easily and efficiently be decoded at any spatial resolution. Second, the embedded bit-stream generated by the Matching Pursuit coder can be adapted to any rate constraints, while the receiver is guaranteed to always get the most energetic components of the MP representation. Most importantly, Matching Pursuit streams offer the advantage of decoupling spatial and rate adaptivity, that can be performed independently. Adaptive decoding is now discussed in more details in the remainder of the section.

7.6.2 Spatial adaptivity

Due to the structured nature of our dictionary, the Matching Pursuit stream provides inherent spatial adaptivity. As explained in Chapter 6, the group law of the similitude group of \mathbb{R}^2 indeed applies [4] and allows for invariance with respect to *isotropic* scaling of α , rotation of Θ and translation of $\vec{\beta}$. Therefore, when the compressed image $\hat{\mathbf{s}}$ is submitted to any combination of these transforms (denoted here by the group element η), the indexes of the MP stream can simply be transformed with help of the group law:

$$\mathcal{U}(\eta)\hat{\mathbf{f}} = \sum_{n=0}^{N-1} \langle \mathbf{g}_{\gamma_n}, \mathbf{R}^n \mathbf{f} \rangle \mathcal{U}(\eta) \mathbf{g}_{\gamma_n} = \sum_{n=0}^{N-1} \langle \mathbf{g}_{\gamma_n}, \mathbf{R}^n \mathbf{s} \rangle \mathcal{U}(\eta \circ \gamma_n) \mathbf{g}. \quad (7.35)$$

In the above expression $\gamma_n = (\mathbf{a}_n, \theta_n, \mathbf{b}_n)$ represents the parameter strings of the atom encoded at iteration n , with scaling \mathbf{a}_n , rotation θ_n and translation \mathbf{b}_n , and $\eta = (\alpha, \Theta, \vec{\beta})$ represents the geometric transformation that is applied to the set of atoms. The decoder can apply the transformations to the encoded bit-stream simply by modifying the parameter strings of the unit-norm atoms, according to the group law of similitude, where

$$(\mathbf{a}, \theta, \mathbf{b}) \circ (\alpha, \Theta, \vec{\beta}) = (\alpha \cdot \mathbf{a}, \theta + \Theta, \mathbf{b} + \alpha \cdot r_{\Theta} \vec{\beta}). \quad (7.36)$$

In other words, if $\eta_\alpha = (\alpha, 0, 0)$ denotes the isotropic scaling by a factor α , the bit-stream of an image of size $W \times H$, after entropy decoding, can be used to build an image at any resolution $\alpha W \times \alpha H$ simply by multiplying positions and scales by the scaling factor α . The coefficients have also to be scaled with the same factor to preserve the energy of the different components. The quantization error on the coefficient will therefore also vary proportionally to the scaling factor, but the absolute error on pixel values will remain almost unchanged, since the atom support also varies. Finally, the scaled image is obtained by:

$$\mathcal{U}(\eta_\alpha)\hat{\mathbf{s}} = \alpha \sum_{n=0}^{N-1} c_{\gamma_n} \mathbf{g}_{\eta_\alpha \circ \gamma_n}. \quad (7.37)$$

The modified atoms $\mathbf{g}_{\eta_\alpha \circ \gamma_n}$ are simply given by Eq. (6.7) to Eq. (6.9), where \mathbf{b} and \mathbf{a} are respectively replaced by $\alpha \mathbf{b}$ and $\alpha \mathbf{a}$. It is worth noting that the scaling factor α can take any positive real value,

as long as the scaling is isotropic. Atoms that become too small after transcoding are discarded. This allows for further bit rate reduction, and avoids aliasing effects when $\alpha < 1$. The smallest atoms generally represent high frequency details in the image, and are located towards the end of the stream. The MP encoder initially sorts atoms along their decreasing order of magnitude, and scaling does not change this original arrangement.

Finally, scaling operations are quite close to image editing applications. The main difference is in the use of the scaling property. Scaling will be used at a server, within intermediate network



(a) $\sqrt{\frac{1}{2}}$



(b) Original size

(c) $\sqrt{2}$

Figure 7.24: *Lenna* image of size 256×256 encoded with MP at 0.3 bpp (b) and decoded with scaling factors of $\sqrt{\frac{1}{2}}$ (a) and $\sqrt{2}$ (c).

nodes, or directly at the client in transcoding operations, while it could be used in the authoring tool for editing. Even in editing, the geometry-based expansion provides an important advantage over conventional downsampling or interpolation functions, since there is no need for designing efficient filters. Other image editing manipulations, such as rotation of the image, or zoom in a region of interest, can easily be implemented following the same principles. The simple spatial adaptation procedure is illustrated in Figure 7.24, where the encoded image of size 256×256 has been re-scaled with irrational factors $\sqrt{\frac{1}{2}}$ and $\sqrt{2}$. The smallest atoms have been discarded in the down-scaled image, without impairing the reconstruction quality. The up-scaled image provides a quite good quality, even if very high-frequency characteristics are obviously missing since they are absent from the initial (compressed) bit-stream.

Table 7.1 shows rate-distortion performance for spatial resizing of the 256×256 *Lenna* image compressed at 0.3 bpp, with the proposed Matching Pursuit coder, and JPEG2000 respectively. It presents the PSNR values of the resized image, as well as the actual bit rate after transcoding. It also shows the PSNR values for encoding directly at the target spatial resolutions, for equivalent rates. The reference images in Table 7.1 have been obtained by resizing the original 512×512 pixel *Lenna* image to 256×256 and to 128×128 with a standard image processing application (Gimp). The reference image 128 DF is an image of size 128×128 , obtained by taking the output of the wavelet Daubechies 9/7 filter bank at this resolution (128×128) when giving as input signal the 256×256 *Lenna* reference image.

It can be seen that our scheme offers results competitive with respect to state-of-the-art coders like JPEG2000 for octave-based downsizing. The quality of the down-scaled images is slightly inferior for the MP coder, but the JPEG2000 transcoded image rate is largely superior to the MP stream one. On the other hand, MP allows for non-dyadic spatial resizing, as well as easy up-scaling. The scaling operation does not significantly affect the quality of the image reconstruction from MP streams. Even in the up-scaling scenario, the transcoded image provides a very good approximation of the encoding at the target (higher) resolution. Next, results are also reported for a 128×128 reference image obtained from the output of the Daubechies 9/7 filter bank, that is used in the JPEG2000 encoder. We can note that the PSNR after transcoding is obviously improved for JPEG2000 in that case, and that the PSNR value for MP is decreased. We could expect the reverse behavior with a reference image obtained with Gaussian filtering: this illustrates the importance of the reference image in spatial resizing performance evaluation. It is finally interesting to note that MP incurs a loss of almost 1.5 dB when coding that particular reference image, mostly due to the aliasing present in this low frequency image*, that the proposed dictionary cannot efficiently capture.

7.6.3 Rate scalability

Matching Pursuit offers an intrinsic multiresolution advantage, which can be efficiently exploited for rate adaptivity. The coefficients are by nature exponentially decreasing so that the stream can simply be truncated at any point to provide a SNR-adaptive bit-stream, while ensuring that the most energetic atoms are kept. The simplest possible rate adaption algorithm that uses the progressive nature of the Matching Pursuit stream works as follows. Assume an image has been encoded at a high target bit-rate R_b , using the rate controller described in Section 7.4. The encoded stream is then restricted to lower bit budgets r_k , $k = 0, \dots, K$ by simply dropping the bits $r_k + 1$ to R_b . This simple rate-adaption, or filtering operation is equivalent to dropping the last iterations in the MP expansion, focusing on the highest energy atoms.

*The Daubechies 9/7 filter gives aliasing when reconstructing at a smaller resolution, aliasing which is compensated for the reconstruction of the original size image by information contained in the higher frequency bands.

Encoder		128x128	128 DF	256x256	512x512
MP	PSNR	28.32	24.62	29.70	27.40
	Rate [bpp]	0.8	0.8	0.3	0.08
	PSNR w/o tr.	29.48	28.03	29.70	28.48
JPEG2000	PSNR	28.90	30.53	30.19	-
	Rate [bpp]	1.02	1.02	0.3	-
	PSNR w/o tr.	33.66	31.08	30.19	-

Table 7.1: Comparison of spatial adaptivity of the MP encoder and JPEG2000. PSNR values are compared to quality obtained without transcoding (w/o tr.). 128 DF is the 128x128 image obtained with the JPEG2000 wavelet filter.

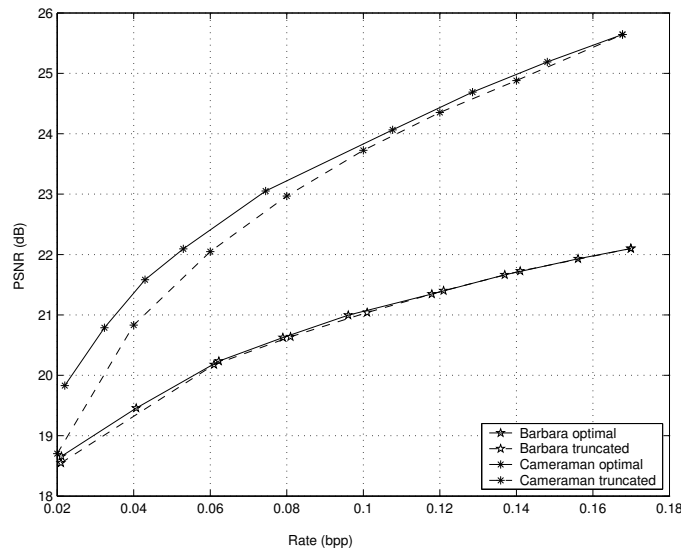


Figure 7.25: Rate-distortion characteristics for MP encoding of the 256x256 *Barbara* and *Camera-man* images at 0.17 bpp, and truncation/decoding at different (smaller) bit rates.

Figure 7.25 illustrates the rate adaptivity performance of the MP encoder. Images have been encoded with MP at a rate of 0.17 bpp and truncated to lower rates r_k . For comparison, the bit-stream has also been encoded directly at the different target rates r_k , as described in Section 7.4. It can be seen that there is a very small loss in PSNR with respect to the optimal MP stream at the same rate. This loss is due to the fact that the rate truncation simply results in dropping iterations, without using the optimal quantizer settings imposed by rates r_k as proposed in Section 7.4.2. The quantization parameters are not optimal anymore with respect to the truncation rate, but the penalty is quite low away from very low coding rates. The loss in performance is larger for images that are easier to code, since the decay of the coefficients is faster. Nevertheless, both optimal and truncated rate-distortion curves are quite close, which shows that a simple rate adaption method, though quite basic, is very efficient.

Next section shows a dequantization scheme, based on the mean energy decay of the coefficients, that improves the performance of this quantizer and reduces the effect of rate truncation.

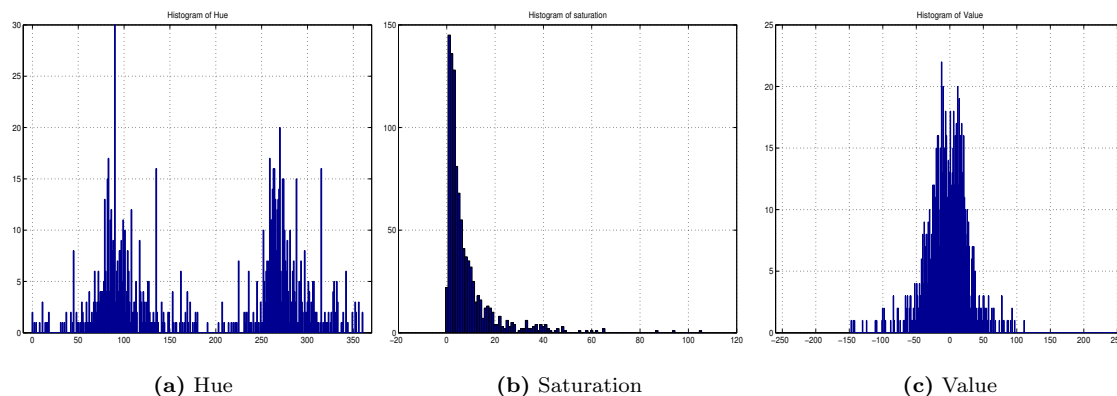


Figure 7.26: Histograms of the MP coefficients when represented in HSV coordinates.

7.7 Color image coding with Matching Pursuit

Now that the complete MP coder for gray-scale image has been presented, the natural question that arises is how color images can be coded. The MP color image coder very naturally follows the same coding scheme than the gray-scale image coder for the atom indices, as these atom indices have exactly the same properties for color images and for gray-scale images. The only difference between color images and gray-scale images in the MP decomposition is the value of the projection of the basis functions: color images have three coefficients (one projection value for every color channel), while gray-scale images have only one.

It has been seen that color images are represented in the RGB color space, in order to take profit from the redundancy among channels. The quantization of the MP coefficients of color images has to be done carefully, and cannot be performed in the RGB color space, because the quantization error would bring color distortion, very annoying for the Human observer. A color space that does not create color distortion in the quantization of color images MP coefficients has to be found. Once this color space has been found, the appropriate quantization scheme for this color space has to be found.

7.7.1 Color Image Parameter coding

Coefficient quantization

A fundamental goal of data compression is to obtain the best possible fidelity for a given data rate or, equivalently, to minimize the rate required for a given fidelity. Due to the structure of the coefficient distribution, centered around the diagonal of the RGB cube, an efficient color quantization strategy is not anymore to code the raw value of the R, G and B components, but instead to code the following parameters: the projection of the coefficients on the diagonal, the distance of the coefficient from the diagonal and the direction where it is located. This is equivalent to coding the MP coefficients in an HSV color space, where V (Value) becomes the projection of RGB coefficients on the diagonal of the cube, S (Saturation) is the distance of the coefficient to the diagonal and H (Hue) is the direction perpendicular to the diagonal where the RGB coefficient is placed. Figure 6.27 shows the 3D distribution of the MP coefficients when their coefficient values are computed as the projection of the basis function in the red, the green and the blue channel and their distribution when these values are transformed to projections in the HSV color space. The HSV values of the

MP coefficients present the following distribution: the Value distribution is Laplacian centered in zero (see Figure 7.26(c)), Saturation presents an exponential distribution (see Figure 7.26(b)), and a Laplacian-like distribution with two peaks can be observed for Hue values (Figure 7.26(a)). Once the HSV coefficients have been calculated from the available RGB coefficients, the quantization of the parameters is performed as follows:

- V is exponentially quantized with Algorithm 7.1. The choice of exponential quantization is driven by the exponential decay of V coefficients as the iteration number increase. The value that will be given as input to the arithmetic coder will be $N_j(l) - Quant(V)$, where $N_j(l)$ is the number of quantization levels that are used for coefficient l .
- H and S are uniformly quantized, since the iteration number does not seem to have any particular influence on their magnitude.

Entropy coding of the parameters

The entropy coding of the quantized parameters is performed through and adaptive arithmetic coder based on [210], with the probability update method used in [54]. The only difference among the entropy coding of color images and gray-scale images is the number of fields to be entropy coded (color images need two extra fields to represent color information). Note that this coder is still under study, and currently does not yet behave optimally. A more elaborated study of the entropy of the parameters, and improved histogram initialization and update methods are necessary to obtain better compression performance. However, the bit-stream obtained is still representative enough of the behavior of the MP coder.

7.7.2 Results

In the previous section it has been shown that the choice of the RGB color space is the most adapted to the MP algorithm which uses the same mother function for the three color channels. All results presented in this section thus use RGB color space for MP expansion.

Figure 7.27 first compares the compression performance of the proposed MP encoder with state-of-the-art JPEG-2000. It can be seen that MP advantageously compares to JPEG-2000, and even performs better at low bit rates. This can be explained by the property of MP to immediately capture most of the signal features in a very few iterations. At higher rates, the advantage of redundant expansion obviously decreases in comparison to orthogonal signal decomposition as performed by JPEG-2000. Note that the PSNR values have been computed in the Lab color space [59], in order to closely evaluate the Human Visual System perception. Figure 7.28 proposes visual comparisons between MP and JPEG-2000, and it can be clearly seen that MP indeed offers better visual quality at low bit rate.

In addition to interesting compression properties, the MP bit-stream offers highly flexible adaptivity in terms of rate and spatial resolution. Figure 7.29 shows the effects of truncating the MP expansion at different number of coefficients. It can be observed that the MP algorithm will first describe the main objects in a sketchy way (keeping the colors) and then it will refine the details. The stream generated by MP thus truly offers an intrinsic rate scalability in image representation.

Finally, MP decompositions with anisotropic refinement atoms are covariant under isotropic dilations, as explained in Chapter 6. This means that the compressed image can be resized with any ratio α , including irrational factors (see Figure 7.30). These spatial scalability properties, added to very low complexity decoder implementation, make MP especially useful for asymmetric applications with heterogeneous end-user terminals.

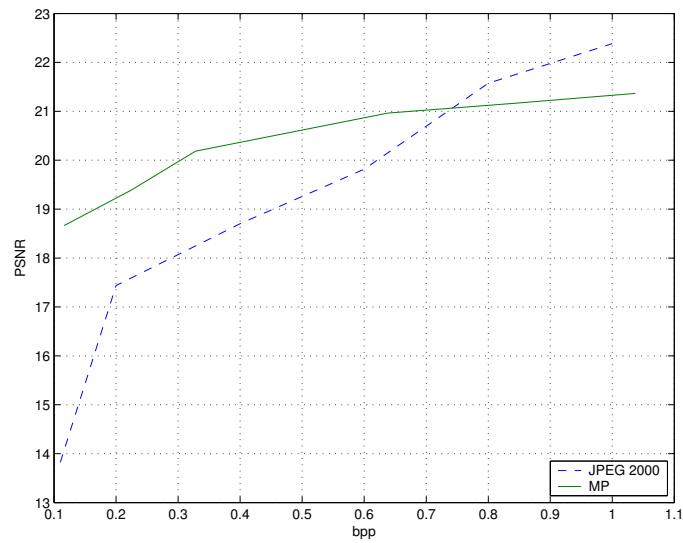


Figure 7.27: PSNR comparison between JPEG 2000 and MP. The PSNR has been computed in the CIELAB color space.

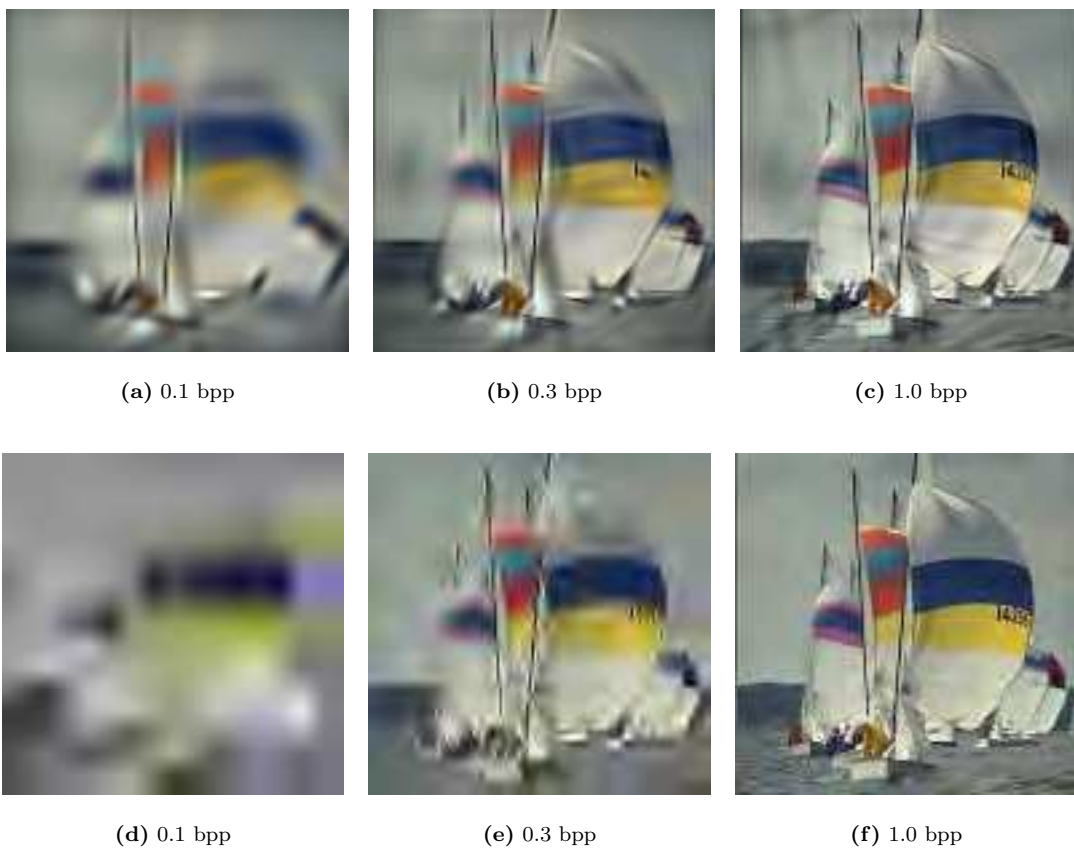


Figure 7.28: Top row, MP of sail with coefficients quantized in HSV color space. Bottom row, JPEG2000 for the same bit-rate.

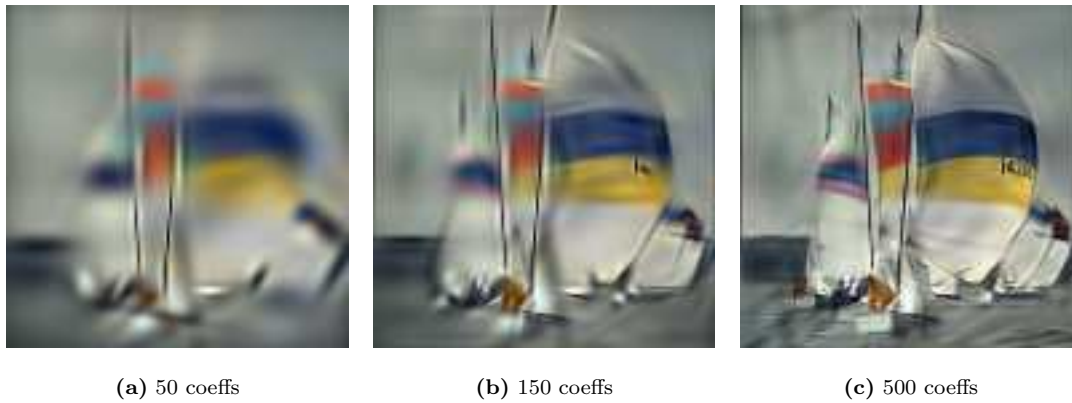


Figure 7.29: MP of sail for 50, 150 and 500 coefficients.

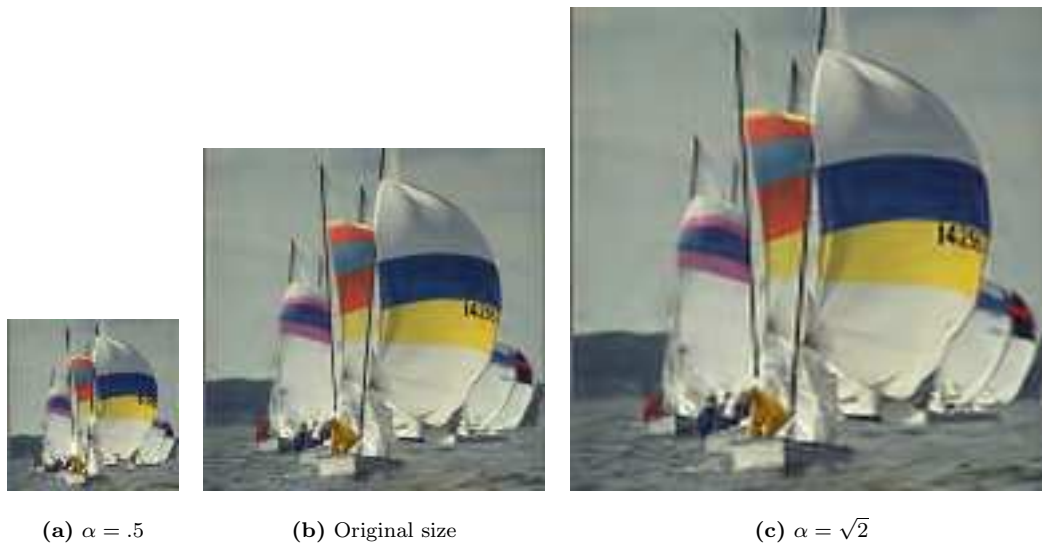


Figure 7.30: Example of spatial scalability with MP streams.

7.8 Conclusions

This chapter has introduced a lossy image compression scheme based on Matching Pursuit. This compression scheme has been introduced for gray-scale and for color images. Both for gray-scale and color images, the coding scheme presented in the scope of this thesis is useful for low to mid bit-rate ranges. It has been seen that for mid bit-rate ranges, the PSNR of the reconstructed images is not higher than the one obtained by the JPEG2000 image coder. Nevertheless, due to the basis functions that have been used, which are similar to the V1 receptive fields of the Human Visual System, the distortion introduced by MP with AR atoms is less annoying to the human observer than the ringing introduced by wavelet like schemes. In addition, the MP based on AR atoms image coder allows for geometrical transformations in the coded domain, which may be useful for visual communications, where the size of the display of the end terminal is variable and (possibly) unknown for the transmitter.

Compression schemes based on redundant dictionaries open interesting horizons for new image compression schemes that are more adapted to the nature of images. These new compression schemes can, not only see the regularity of contours and thus use less coefficients to represent them, but help understand the nature of some image components.

The coder presented in this thesis is in an experimental stage, but it has already shown interesting results. With this, it can be deduced that further investigation and development of redundant image or video representation and approximation schemes may be of interest, and give new breakthroughs in image coding and representation.

The main drawback of this coder is its low efficiency at high bit-rates, due to the nature of the Matching Pursuit algorithm and the redundant dictionary. This drawback could be solved by the use of hybrid coding schemes. Of course, the use of hybrid coding schemes would limit the capabilities of the coder to perform affine image transformations in the coded domain.

Conclusions

8.1 Summary

This thesis has pointed out, in the introduction, the need for efficient or *sparse* approximation of images and other multidimensional geometrically structured data. In the particular domain of image processing, sparse and flexible image approximations or representations can bring solutions to new image processing challenges. Some of these challenges include representations that provide a good image modeling and parametrization. This may contribute to having better descriptions of the image content, and may give good performances on data mining or automatic database indexation. In this sense, this thesis focuses on the study of sparse image approximations, and their applications to flexible image coding.

In order to better understand the nature of visual information and how it can be sparsely represented, Chapter 2 has reviewed different sources that model and capture images. To achieve sparse signal approximations, inspiration from nature, with an overview of the Human Visual System was proposed. Human Visual System uses anisotropic and oriented basis functions, and it is supposed to perform a sparse approximation of the visual information. This is the first hint that leads toward the use of anisotropic and oriented basis functions to achieve sparse image approximations. However, a more objective analysis is required, to understand why such properties in the basis functions are required.

Chapter 3 used the Horizon Model introduced in Chapter 2 to check mathematically which are the coding and approximation gains when using contour adaptive arbitrary anisotropic and oriented basis functions. This chapter showed that, if a smart representation scheme is used, anisotropy and rotation in the basis function does improve both the approximation properties and the coding performances.

Arbitrary anisotropy and orientation in the basis functions can only be achieved by the use of redundant dictionaries. However, redundant dictionaries do not provide a unique signal representation/approximation. From all the possible approximations of a signal in a redundant dictionary, we would like to recover the sparsest (the one that, for a given number of terms, provides a minimum error, or that, for a given error, has a minimum number of terms in the decomposition).

Chapter 4 enumerates different algorithms and techniques to recover sparse signal approximations from redundant dictionaries and analyzed their performances. From all the algorithms presented in this chapter, Matching Pursuit (MP) seemed to be the most appropriate for dealing with highly redundant geometrical dictionaries for several reasons. The first reason is that MP is flexible and easy to implement, and its approximation error converges to zero for any dictionary of functions. The second reason is that MP, thanks to the fact that it is an iterative algorithm, permits to easily select the number of terms involved in an approximation. Other algorithms, such as Basis Pursuit Denoising, do not allow to choose an exact number of terms, one has to play with some algorithm parameters to reduce or increase the number of terms that are found in order to have a certain amount of non-zero coefficients in a given signal approximation. Finally, Matching Pursuit is able to find, under certain circumstances, the sparsest existing solution in a redundant dictionary. Nevertheless, MP does not guarantee to retrieve the sparsest image approximation for all dictionaries and signals. This motivated a detailed analysis of sparse signal approximations, which is presented in Chapter 5. This analysis of the properties of sparse approximations has, as starting point, the conditions needed for a sparse approximation to be unique and optimal. From these conditions, a test to check the optimality of a sparse approximation was derived. This test has the particularity of being algorithm independent, and tighter than existing ones.

Chapter 6 took the features of the basis functions that have arisen from Chapter 3 to create a redundant dictionary having these features. This dictionary is formed by anisotropic refinement atoms and Gaussians. This chapter presented then an efficient Full Search Matching Pursuit algorithm to perform the image decomposition in such a dictionary. The same chapter emphasizes the fact that the dictionary formed by anisotropic refinement atoms and Gaussians, even though it is able to perform sparser image approximations than other sets of basis functions, fails to efficiently approximate textures. In order to efficiently approximate images with textures, a new dictionary, based on Gabor atoms, is added to the dictionary based on anisotropic refinement atoms. The macro-dictionary formed by the union of the Gaussian dictionary, the Gabor dictionary and the Anisotropic refinement atom dictionary becomes far too redundant for recovering the sparsest signal approximation. To help MP choose the good kind of basis function (Gabor for textures, anisotropic refinement atoms for edges or Gaussians for smooth parts), a Weighted MP (WMP) algorithm is introduced. This WMP algorithm influences the choice of MP by pondering the scalar product of the different basis functions in the dictionary according to *a priori* information of the nature of the signal to represent. Even though the weighted MP algorithm has been introduced to help MP to choose between three different kinds of basis functions, the same scheme can be used to force MP to put more basis functions in image zones where there is more visual information (such as faces in natural images) or to choose image parts that are more important first (for example, for medical images, the part of the image that shows a certain pathology) to be represented before other less significant information. Finally, this chapter explains an adaption of the MP image approximation algorithm to color images through a Vector MP. Interestingly, this color image approximation algorithm does not try to uncorrelate the three different color channels, but profits from the existing correlation among channels.

Finally, Chapter 7 proposed a flexible image coding scheme, that uses the image approximations obtained in Chapter 6 in the anisotropic refinement atoms and Gaussian dictionary. This chapter presents a detailed study about the coding performances of such a dictionary, which evidences the fact that coefficient quantization appears to be of crucial importance in the design of a Matching Pursuit based coding scheme. Thus, an enhanced MP coefficient quantization scheme is designed. This quantization scheme is based on the theoretical upper-bound of the energy of the coefficients and empirical observations of the distribution of the energy of this coefficients and its evolution

with the iteration number. This coding scheme has two main interesting characteristics: it has rate scalability and it allows for several affine image transformations directly in the coded domain. The drawback of such a coding scheme is its incapability of achieving high bit-rates or lossless image representation. However, this drawback can be solved by using hybrid image coding schemes, that combine redundant representations with orthonormal basis.

8.2 Future research

There are more and more techniques that claim to build sparse image approximations. However, obtaining sparse image approximations that allow for image manipulations, such as affine image transformations in the coded domain, is still an open problem. This thesis has presented a possible primary approach to solve this kind of problems, but it has not presented the final solution to these questions. The study of further redundant dictionaries, that still keep geometrical properties in the basis functions but that are even more adapted to natural image sparse approximation is one of the future research lines that come out of this work. These basis functions could have special features, such as low coherence among them or at least block incoherence to facilitate the search of the sparsest approximation.

Signal representation in structured redundant dictionaries has proved to give interesting results, but theoretical bounds for recoverability of the sparsest approximation do not demonstrate that this structured dictionaries have better approximation properties or can more easily recover the sparsest signal approximation than non-structured ones. A study about specific sparsity properties of structured dictionaries can be useful to better understand the properties of such signal decompositions.

This thesis has shown the study of a geometric dictionary to approximate images. This dictionary provides a continuous image representation, that is then sampled to obtain the digital signal. The fact of having this continuous image representation allows for several interesting operations in the coded domain, such as rotation or resizing of the image. Further studies can be of interest, in order to use this continuous version of the image to provide super-resolution images.

Image coding through a Matching Pursuit in a geometric edge adaptive dictionary has provided interesting results for low and medium bit-rates. In addition, due to the fact that MP with AR atoms produces sketchy versions of the image and not ringing artifacts, the low-rate images obtained with MP have generally better visual quality than the low-rate images obtained with wavelets, even for higher wavelet PSNRs. The MP-based image coder has mainly two drawbacks: the loss of compression ratio at high bit-rates and the complexity of the algorithm. MP loses compression ratio at high bit-rates due to the redundancy of the dictionary. In addition, MP cannot achieve lossless image representations. Hybrid schemes, that use MP decompositions and orthogonal basis should be studied, in order to achieve higher bit-rates without losing R-D performances [144]. About the complexity of MP, optimized fast MP schemes can be used, such as tree search schemes [100], in order to speed up the search, but a lot remains to be investigated about this suboptimal approaches.

Another interesting future work is to study the real R-D behavior of image approximations obtained with the AR dictionary. The R-D study performed with the anisotropic quadtree was useful to check whether the inclusion of anisotropy and orientation in the basis function brings any improvement in image approximations. Nevertheless, it does not model the real R-D behavior of image approximations obtained with the AR dictionary. The main reason for this is that the basis functions that approximate edges in this dictionary are not necessarily nested, and thus, the coding scheme that took profit from the nested partitions cannot be used.

The use of redundant structured dictionaries for higher dimensionality data can be very useful

for several applications. The first extension of the work is to apply it in 3D, for example for video signals or for 3D still images [37]. In order to do this, a study about the features the dictionary should have for representing this higher dimensionality data has to be performed.

Finally, one could think about substituting the MP algorithm for some other more sophisticated algorithm, that allows to find the sparsest approximation in more cases than MP does.

APPENDIX

Appendix A

Some proves on the algorithm independent sparsity measures

A.1 Sparsity measures

For the sake of completeness we include some properties of sub-additive (resp. admissible) sparsity measures.

Lemma A.1 ([87, Prop. 1]) *Every admissible sparsity measure is sub-additive, but the reciprocal is false.*

Proof. For $t, u \geq 0$, since $f(t+u)/(t+u) \leq f(u)/u$ and $f(t+u)/(t+u) \leq f(t)/t$, we have

$$f(t) + f(u) \geq f(t+u) \cdot \frac{t}{t+u} + f(t+u) \cdot \frac{u}{t+u} = f(t+u).$$

Now, denoting $\lfloor t \rfloor$ the largest integer such that $\lfloor t \rfloor < t \leq \lfloor t \rfloor + 1$, and f the function such that $f(0) = 0$ and $f(t) = 1 + \lfloor t \rfloor$ for $t > 0$, we let the reader check that f is sub-additive but not admissible.

Lemma A.2 *If f, g are sub-additive sparsity measures, then $f \circ g$ is also a sub-additive sparsity measure.*

Proof. First, since $f(0) = g(0) = 0$ and f, g are non decreasing, $f \circ g$ has the same properties. Now, for any $t, u \geq 0$ we have

$$f[g(t+u)] \leq f[g(t) + g(u)] \leq f[g(t)] + f[g(u)]$$

Figure A.1 illustrates the fact that admissible sparsity measures are not necessarily as “nice” as one could imagine from their most natural examples the ℓ^τ measures. Using the fact that the min or the max of two admissible sparsity measures yields another admissible sparsity measure [87], we can combine the ℓ^0 and the ℓ^1 norms into the “exotic” measure $f(t) := \min(f_1(t), \max(f_0(t), f_1(t)/2))$.

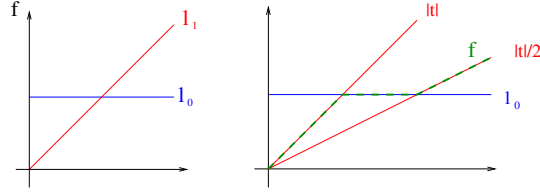


Figure A.1: (a) The “nice” functions $f_0(t)$ and $f_1(t)$ corresponding to the ℓ^0 and the ℓ^1 norm (b) An “exotic” sparsity measure, based on a mix of the ℓ^0 and ℓ^1 measures plotted in (a).

A.2 Proof of Lemma 5.1

For any index set I , we let $C(I) = \{c, \text{support}(c) = I\}$. Since $C_M = \bigcup_{\text{card}(I)=M} C(I)$ we have

$$E_f(C_M) = \bigcup_{\text{card}(I) \leq M} E_f(C(I)). \quad (\text{A.1})$$

By definition, $\delta \in E_f(C(I))$ if, and only if, for all $z \in C(I)$

$$\begin{aligned} \sum_{k \in I} f(|z_k|) &= \|z\|_f \geq \|z + \delta\|_f \\ &= \sum_{k \in I} f(|z_k + \delta_k|) + \sum_{k \notin I} f(|\delta_k|), \end{aligned}$$

or in other words if, and only if, for all $z \in C(I)$

$$\sum_{k \in I} [f(|\delta_k|) + f(|z_k|) - f(|z_k + \delta_k|)] \geq \|\delta\|_f. \quad (\text{A.2})$$

Letting $G_f(t) := \sup_{u \in \mathbb{R}} f(|u|) - f(|u + t|)$ we have

- if $\sum_{k \in I} [f(|\delta_k|) + G_f(\delta_k)] > \|\delta\|_f$ then $\delta \in E_f(C(I))$;
- if $\sum_{k \in I} [f(|\delta_k|) + G_f(\delta_k)] < \|\delta\|_f$ then $\delta \notin E_f(C(I))$.

and the case $\sum_{k \in I} [f(|\delta_k|) + G_f(\delta_k)] = \|\delta\|_f$ depends whether the supremum is achieved in the definition of G_f . For any f one can see that $G_f(t) \geq f(|t|)$ by letting $u = -t$ in the definition of G_f . When f is non-decreasing and sub-additive, we have for all u and t

$$\begin{aligned} f(|u|) - f(|u + t|) &= f(|u + t - t|) - f(|u + t|) \\ &\leq f(|u + t| + |t|) - f(|u + t|) \\ &\leq f(|u + t|) + f(|t|) - f(|u + t|) \\ &= f(|t|) \end{aligned}$$

which shows that we have indeed $G_f(t) = f(|t|)$, and the supremum is achieved in the definition of G_f . Thus, if f is non-decreasing and sub-additive, we have $\delta \in E_f(C(I))$ if, and only if, $\sum_{k \in I} f(|\delta_k|) \geq \|\delta\|_f/2$. From the characterization (A.1) we conclude that $\delta \in E_f(C_M)$ iff

$$\sum_{k \in I_M(\delta)} f(|\delta_k|) \geq \frac{\|\delta\|_f}{2}$$

where $I_M(\delta)$ indexes the M largest components $|\delta_k|$.

Appendix B

Computation of the sampling of the dictionary parameters

Chapter 6 joined three dictionaries to make a macro-dictionary (or multi-component dictionary). The PSNR of MP image approximations improves with the number of components of the “multi-component” dictionary. This is because the MP algorithm has a richer variety of atoms available to catch the structures of the image. In the following sections, the different dictionaries used in this work are described. Each one of the basic dictionaries used to build the multi-component dictionary is based on a generic, continuous domain, mother function; each mother function is tuned to a variety of geometric transformations to generate each dictionary. For compression purposes, the maximum amount of possible geometric transformations has to be controlled. Indeed, the parameters that conform there is a need of sampling the parameters that define the basis functions. One can find in the following how this sampling is performed, and the effect it has on each one of the sub-dictionaries used in this thesis.

B.1 Dictionary \mathcal{D}_{AR} based on AR atoms

The use of Anisotropic Refinement (AR) dictionaries was introduced by [195] to code image edges more efficiently. Furthermore, it has been shown in [139], [64] and [66] that Anisotropic Refinement atoms are more suitable for edge image coding. Moreover, Human Visual System (HVS) is adapted to this type of functions [138]. Hence, this type of atoms seems to be an appropriate choice to code the edges of images.

An AR atom is defined as a combination of a Gaussian in an axis and its second derivative in the other axis

$$g_\gamma(x, y) = K(4x^2 - 2)e^{-(x^2 + y^2)}, \quad (\text{B.1})$$

where K is a normalization constant to have unitary norm. Assuming that our image has a size of $N_x \times N_y$ pixels, we can define the set of parameters $\gamma = (\mathbf{p}, \mathbf{s}, \theta)$ necessary to generate the whole dictionary:

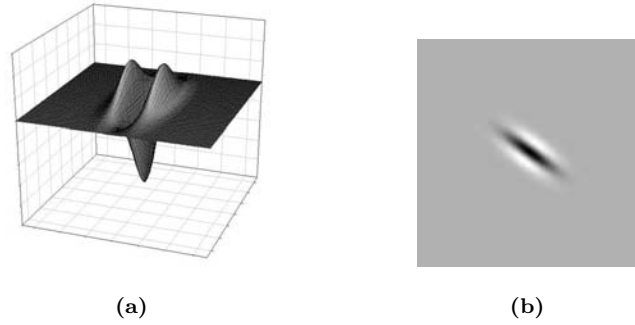


Figure B.1: Anisotropic Refined atom.

$$\begin{array}{ll}
 \mathbf{p} = [p_x, p_y] \text{ where } p_x \in [0, N_x), p_y \in [0, N_y) & \text{translations} \\
 \mathbf{s} = [s_x, s_y] & \text{scaling factors} \\
 \theta \in [0, \pi) & \text{rotation,}
 \end{array}$$

with \mathbf{p} the translation vector that will set the atom location within the image, \mathbf{s} where s_x is the dilation in the x axis and s_y is the dilation in the y axis and θ the rotation angle.

At this point, we have to choose our parameters carefully such that a large enough variety of orientations, scalings and translations is available to efficiently represent edges in images. If $\Delta\theta$ and Δs are small enough finite linear expansions of space-frequency atoms are dense in $\mathbf{L}^2(\mathbb{R}^2)$, hence this dictionary is also complete and, then, valid for image approximation. To satisfy these conditions we have chosen our parameters in this way:

- $\Delta\theta = \frac{\pi}{18}$.
- $\{s_x, s_y\} \in [0, NN \cdot (\lfloor \log_2(N) \rfloor - 3)] \in \mathbb{Z}$ where $NN \in [1, \log_2(N)] \in \mathbb{Z}$, $N = \min(N_x, N_y)$ and $s_x > s_y$.

Once we have defined the parameters to generate the atoms family, we can define the procedure to create the atom. This procedure is non-commutative, so, the order is fixed and its application is:

1. Apply the translation by $[p_x, p_y] \in \mathbb{Z}^2$.
2. Rotate by θ the translated atom.
3. Scale the translated and rotated atom by $\sigma_x = 2^{\frac{s_x}{NN}}$ and $\sigma_y = 2^{\frac{s_y}{NN}}$ in the axis x and y respectively.

This leads to compute:

$$x_\gamma = \frac{(x - p_x) \cos(\theta) + (y - p_y) \sin(\theta)}{2^{\frac{s_x}{NN}}} \quad (\text{B.2})$$

$$y_\gamma = \frac{(x - p_x) \sin(\theta) - (y - p_y) \cos(\theta)}{2^{\frac{s_y}{NN}}}, \quad (\text{B.3})$$

and $g_\gamma = g(x_\gamma, y_\gamma)$.

B.1.1 A Typical Parameter Discretization

Translations The translations must cover the whole image, hence:

$$[0, 0] \leq [p_x, p_y] \leq [N_x - 1, N_y - 1]. \quad (\text{B.4})$$

NN The quality of the approximated image increases with NN , but the size of the dictionary too. If NN is too small some artifacts (lines) may appear in the image, and if NN is too large, the computational load to perform Matching Pursuit becomes prohibitive. A good compromise is to take

$$NN = 3. \quad (\text{B.5})$$

Scaling Factors The scaling factors are chosen in a way that the smaller scale is the unit and the bigger is proportional to the size of the image. So:

$$2^{\frac{s_{i\min}}{NN}} = 1 \quad \rightsquigarrow \quad s_i \geq 0 \quad (\text{B.6})$$

$$2^{\frac{s_{i\max}}{NN}} = \min(N_x, N_y) \quad \rightsquigarrow \quad s_i \leq NN \log_2(\min(N_x, N_y)), \quad (\text{B.7})$$

where $i = x, y$ and

$$s_i \in [0, NN \cdot (\lfloor \log_2(N) \rfloor - 3)], \quad (\text{B.8})$$

where the factor 3 has been added in order limit the biggest scale used (bigger ones are rarely used in natural images).

Rotation A the rotation is in steps of $\frac{\pi}{18}$ degrees. Rotations, in order to cover all the spectrum, must range from 0 to π radians (because all the atoms are real and have a spectral component in the positive frequency plain and its symmetric component in the negative plain. Otherwise, it should move from 0 to 2π). So,

$$\text{phase}_n = n\Delta\theta = n\frac{\pi}{18}. \quad (\text{B.9})$$

Size of the dictionary With the former assumptions, the size of the dictionary can be calculated: 26.836.992 atoms for an image of 128 squared pixels.

B.2 Dictionary \mathcal{D}_G based on Gaussian atoms

The use of a dictionary based on Gaussian functions has the purpose of better approximating smooth image parts. The AR atoms are suitable for approximating contours, but they are not adapted to the low frequency band of natural images. In previous works ([66],[65]), the lower frequencies of an image were represented by either using a downsampled copy of the image or by just leaving this task to the AR atoms. The drawback of using a downsampled image to code the baseband is that it limits coding flexibility.

A Gaussian atom is defined as:

$$g_\gamma(x, y) = \sqrt{2}Ke^{-(x^2+y^2)}, \quad (\text{B.10})$$

where γ is the set of parameters to generate the dictionary and K a constant that normalizes the discrete norm of g_γ . Assuming our Gaussian atoms isotropic, we can define our parameters x_γ and y_γ :

$$x_\tau = \frac{(x - p_x)}{\sigma_x} = \frac{(x - p_x)}{2^{\frac{s_x}{NN}}} \quad (\text{B.11})$$

$$y_\tau = \frac{(y - p_y)}{\sigma_y} = \frac{(y - p_y)}{2^{\frac{s_y}{NN}}}. \quad (\text{B.12})$$

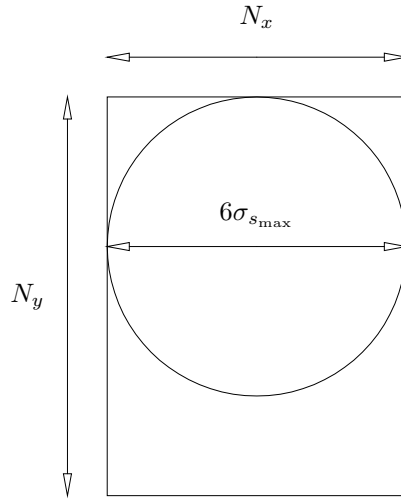


Figure B.2: Framework to calculate $\sigma_{s_{\max}}$

where $s_x = s_y = s$. The design of the dictionary must be carefully done in order to cover a variety of baseband areas in the Fourier domain. Hence, we must impose few constrains on the s parameter but, for simplicity, let's call $2^{\frac{s}{N}} = \sigma_s$. We call σ_s due that this parameter refers to the *spatial* σ in contraposition with the *frequency* σ_f . We have to review that the Fourier transform of a Gaussian is also a Gaussian and their σ 's are connected by:

$$\sigma_s \cdot \sigma_f = 2. \quad (\text{B.13})$$

Applying few constraints we might find the upper and lower bounds of σ_s :

- $\sigma_{s_{\max}} \rightsquigarrow$ The biggest Gaussian we will be able to put in our image (in spatial sense) will be determined by the image itself: its size (taking the smallest side of an $N_x \times N_y$ pixels image). We know that the 99% of the energy of the Gaussian is concentrated in the $6\sigma_s$ area of it. This leads us to:

$$6 \cdot \sigma_{s_{\max}} = \min(N_x, N_y) \rightsquigarrow \sigma_{s_{\max}} = \frac{\min(N_x, N_y)}{6}. \quad (\text{B.14})$$

This Gaussian will have the lowest frequency representation, so by applying B.13 we find the smallest bandwidth our Gaussians will cover:

$$BW_{\min} = \sigma_f \cdot 3 = \frac{2}{\sigma_s} \cdot 3 = \frac{36}{\min(N_x, N_y)}. \quad (\text{B.15})$$

- $\sigma_{s_{\min}} \rightsquigarrow$ The previous constrain was found in the spatial domain of the image; this other is found in the Fourier domain. As we want to cover the baseband of the Fourier domain, we have already defined the constrain that will give the lowest frequency; to find the upper bound we have to find the point where our Gaussians collide with the lowest central frequency of the AR dictionary atoms. We have to find the frequency (ω_x, ω_y) of the atom with lowest frequency where it has maximum power. First, let's find the Fourier transform of a generic AR atom:

$$G_\lambda(\omega_x, \omega_y) = -\pi \sigma_x \omega_x^2 e^{-\frac{(\sigma_x \omega_x)^2 + (\sigma_y \omega_y)^2}{4}}, \quad (\text{B.16})$$

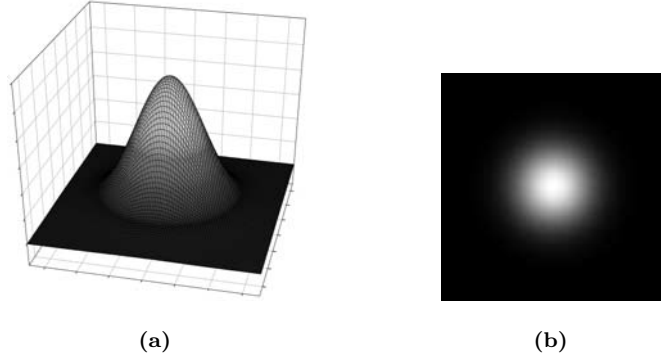


Figure B.3: Gaussian atom.

where $\sigma_x = 2^{\frac{s_x}{N}}$ and $\sigma_y = 2^{\frac{s_y}{N}}$ are the scaling factors of the AR atom as we defined in B.2 and B.3. To find the maximum of this function we can impose the condition $\omega_y = 0$ due to the symmetry of our spectrum. So, we have:

$$\frac{\partial G_\lambda(\omega_x, 0)}{\partial \omega_x} = -2\pi\sigma_x\omega_x e^{-\frac{(\sigma_x\omega_x)^2 + (\sigma_y\omega_y)^2}{4}} \left(1 - \frac{(\omega_x\sigma_x)^2}{4}\right) = 0. \quad (\text{B.17})$$

Avoiding the trivial solution $\omega_x = 0$, it leads to find the lowest frequency $\omega_{x_{\min}}$ of the AR atoms (so that is when the scaling σ_x factor is maximum):

$$\omega_{x_{\min}} = \frac{2}{\sigma_{x_{\max}}}. \quad (\text{B.18})$$

Also this lowest frequency of AR atoms is the biggest bandwidth our Gaussian dictionary will represent, that is

$$BW_{\max} = \frac{2}{\sigma_{x_{\max}}}. \quad (\text{B.19})$$

Once we have set this lowest frequency, we can find the lower bound of σ_s . Taking in consideration:

$$3\sigma_f = \frac{2}{\sigma_{x_{\max}}}. \quad (\text{B.20})$$

Then, by applying B.13 we find:

$$\sigma_{s_{\min}} = 3 \cdot \sigma_{x_{\max}}. \quad (\text{B.21})$$

B.2.1 A Typical Parameter Discretization

The design parameters chosen for this Gaussian dictionary are:

Translation We will make the same assumptions done in the previous dictionary.

Scales By strictly applying the relations B.14 and B.21 we find that (in concrete for our squared images of 128 pixels):

$$\sigma_{s_{\max}} = \frac{128}{6} \approx 21 \quad (\text{B.22})$$

$$\sigma_{s_{\min}} = 3\sigma_{AR_{\max}} = 3 \cdot 2^{\frac{12}{3}} = 48. \quad (\text{B.23})$$

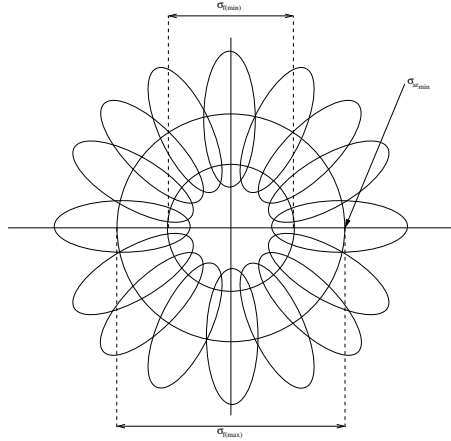


Figure B.4: Gaussian design frequency scheme. The bigger circle corresponds to the spectrum of the Gaussian with $\sigma_{s_{min}}$ (and $\sigma_{f_{max}}$ inversely) defined by the lowest frequency of the AR Atoms. The small circle corresponds to the $\sigma_{s_{min}}$ (and $\sigma_{f_{min}}$ accordingly) defined by the size of the image.

The reader can see that these bound for σ_s are not logical in the sense that $\sigma_{s_{min}} > \sigma_{s_{max}}$. The reason is because our Anisotropic Refinement dictionary has too big scales. There would be two solutions: to reduce the scales of the AR dictionary or relax our assumptions in the calculation of $\sigma_{s_{min}}$. The first option leads to reduce the AR dictionary and the following results are poor because Matching Pursuit does not have enough atoms to catch efficiently the structures of the image. The second option seems more feasible, thus, we will relax the constrain and, instead of taking the maximum $\sigma_{AR_{max}}$ we would take the first σ'_{AR} that accomplishes the criteria. Hence,

$$\sigma_{s_{min}} = 3\sigma'_{AR} = 3 \cdot 2^{\frac{8}{3}} \approx 19, \quad (\text{B.24})$$

and, finally, $\sigma_s \in [19, 21]$.

Size of the dictionary With the former assumptions, the size of the dictionary can be calculated: 49.152 atoms for an image of 128 squared pixels.

B.3 Dictionary \mathcal{D}_{Ga} based on Gabor Atoms

In most of the previous research done on Matching Pursuit over images, most of the decompositions have been done over dictionaries based on Gabor atoms [97] due to its good time-frequency localization. For our purposes, the use of Gabor atoms is justified due to the fact that they are appropriate functions to code patterns and textures.

Note that we will take isotropic Gaussians instead of anisotropic Gaussians here. The only new constrain here is the election of the frequencies ω_x ; we must choose the frequencies carefully to accomplish the Nyquist criteria and set an overlap ratio that would guarantee the density condition in the associated Hilbert space.

Then for each scaling factor $\sigma_s = 2^{\frac{s}{N}}$ we must calculate the set of frequencies ω_{x_i} it will represent. For our purpose we can define a set of constrains that will determinate those frequencies:

Condition 1: It is clear that a determined Gaussian will be able to represent a range of (normalized) frequencies $[\omega_{x_{min}}, 1]$ where $\omega_{x_{min}}$ is the minimum frequency. This minimum frequency will the

determined by the size of the Gaussian, so:

$$6\sigma_s > T \rightsquigarrow \omega_x > \frac{1}{6\sigma_s}, \quad (\text{B.25})$$

where T is the spatial period of the atoms. We can rewrite this equation in term of the scaling factor s :

$$\omega_x > \frac{2^{-\frac{s}{NN}}}{6}. \quad (\text{B.26})$$

Also, sometimes it is not critical to reach the lower bound since low frequencies would be already represented by either the Anisotropic refinement atoms or the Gaussian atoms. So, it is possible to relax the lower frequency condition by applying a *correction factor*:

$$6\sigma_s > \alpha T \rightsquigarrow \omega_x > \frac{\alpha}{6\sigma_s}. \quad (\text{B.27})$$

Then the corrected lower frequency is:

$$\omega_x > \alpha \frac{2^{-\frac{s}{NN}}}{6}. \quad (\text{B.28})$$

Remember that we are here working with normalized frequencies. If we want to work with absolute frequencies we have to multiply by the sampling frequency, that is $\frac{N_x}{2}$.

Condition 2: Once we have set the range of frequencies, we must define the step between two represented frequencies in this set. That is the increment between two adjacent frequencies. We could represent *all* the frequencies in the range $[\omega_{x_{\min}}, 1]$ incrementing by unary steps $\frac{2}{N_x}$. But this would be a waste of resources. Indeed, according with Figure B.5, there would be too much overlapping. We can overlap less, for example just half a Gaussian in the Fourier domain:

$$\Delta f = 3\sigma_f. \quad (\text{B.29})$$

Or, rewritten into terms of s :

$$\Delta f = 6 \cdot 2^{-\frac{s}{NN}}. \quad (\text{B.30})$$

Also, we can control the overlap:

$$\Delta f' = \beta \Delta f \quad \beta \in (1, 2]. \quad (\text{B.31})$$

Then, finally, we can generate the associated frequencies per each scaling factor s :

$$\omega_{x_i}(s, NN) = 1 - i\Delta f'(s, NN) \quad (\text{B.32})$$

$$i \in [0, j \text{ t.q. } \omega_{x_j} > \omega_{x_{\min}}]. \quad (\text{B.33})$$

B.3.1 A typical parameter discretization

The design parameters chosen for the Gabor dictionary are:

Translation We make the same assumptions done in the previous dictionaries.

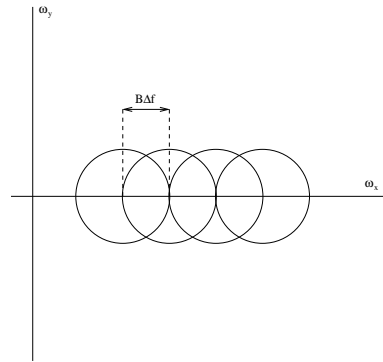


Figure B.5: Incremental factor.

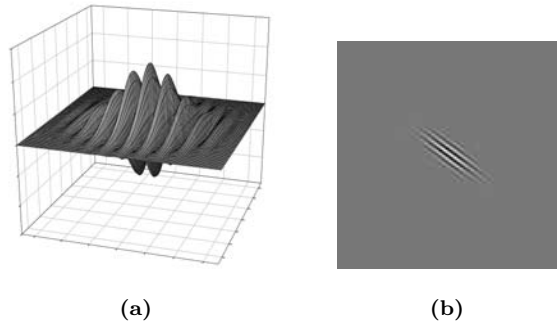


Figure B.6: Gabor atom.

Scaling A good compromise is:

$$s_i = \left[\frac{NN \cdot (\lfloor \log_2(N) \rfloor - 3)}{2}, NN \cdot (\lfloor \log_2(N) \rfloor - 3) \right] \quad (\text{B.34})$$

Correction parameters The adjustable parameters that defines our dictionary have been set as:

$$\alpha = 1.5 \quad \beta = 1 \quad (\text{B.35})$$

Size of the dictionary With the former assumptions, the size of the dictionary can be calculated: 44.236.800 atoms for an image of 128 squared pixels.

Bibliography

- [1] (2003). Cool optical illusions. <http://www.coolopticalillusions.com/>.
- [2] (2004). *Information technology - JPEG2000 image coding system*. International Organization for Standardization, ISO/IEC 15444 edn.
- [3] N. Ahmed, T. Natarajan, K. R. Rao (1974). Discrete cosine transform. *IEEE Transactions on Computers* **23**(1):90–93.
- [4] J.-P. Antoine, R. Murenzi, P. Vandergheynst (1999). Directional wavelets revisited: Cauchy wavelets and symmetry detection in patterns. *Appl. Comput. Harmon. Anal.* **6**(3):314–345.
- [5] G. B. Arfken, H. J. Weber (1995). *Mathematical Methods for Physicists*, chap. 9.3 Gram-Schmidt Orthogonalization, pp. 558–564. Academic Press.
- [6] R. Baraniuk (2004). Multiscale geometric analysis for higher-dimensional signals. lecture 1: Wedgelets. IHP Summer School 2004 Slide support.
- [7] F. Bergeaud, S. Mallat (1995). Matching pursuit of images. In *Proc. ICIP '95*, vol. 1, pp. 53–56.
- [8] D. Bernier, K. Taylor (1996). Wavelets from square-integrable representations. *SIAM J. on Math. Anal.* **27**(2):594–608.
- [9] D. P. Bertsekas (2004). *Nonlinear Programming*. Athena Scientific, USA, 2 edn.
- [10] M. J. Black, G. Sapiro (1999). Edges as outliers: Anisotropic smoothing using local image statistics. In *Scale-Space Theories in Computer Vision*, pp. 259–270, Second Int. Conf., Scale-Space '99, Springer.
- [11] K. R. Boff, L. Kaufman, J. P. Thomas (1986). *Handbook of perception and Human Performance*. A Wiley-Interscience publication.
- [12] G. M. Boynton, J. Hegdé (2004). Visual cortex: The continuing puzzle of area v2. *Current Biology* **14**(13):R523–524.
- [13] R. Bracewell (1965). *The Fourier Transform and its Applications*. McGraw Hill.
- [14] R. W. Buccigrossi, E. P. Simoncelli (1999). Image compression via joint statistical characterization in the wavelet domain. *IEEE Transactions on Image Processing* **8**(12):1688–1700.
- [15] E. J. Candès, D. L. Donoho (1999). Ridgelets: a key to higher dimensional intermitency? *Philosophical Transactions of the Royal Society of London A* **357**(1760):2495–2509.

-
- [16] E. J. Candès, D. L. Donoho (2000). Curvelets - a surprisingly effective nonadaptive representation for objects with edges. In A. Cohen, C. Rabut, L. Schmaker (eds.), *Curves and Surface fitting, Saint-Malo 1999*, Vanderbilt University Press., Nashville.
- [17] E. J. Candès, F. Guo (2002). New multiscale transforms, minimum total variation synthesis: Applications to edge-preserving image reconstruction. *Signal Processing* **82**(11):1519–1543.
- [18] J. Canny (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(6):679 – 698.
- [19] C. Canton Ferrer, R. M. Figueras i Ventura, P. Vandergheynst (2003). *Texture and edge adaptive Weak Matching Pursuit*. Master’s thesis, EPFL-UPC.
- [20] T. F. Chan, H. M. Zhou (2001). Adaptive ENO-wavelet transforms for discontinuous functions. In T. Chan, T. Kako, H. Kawarada, O. Pironneau (eds.), *12th International Conference on Domain Decomposition Methods*, pp. 93–100.
- [21] S. S. Chen, D. L. Donoho, M. A. Saunders (1998). Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing* **20**(1):33–61.
- [22] S.-S. Chen, C.-B. Yang, K.-S. Huang (2002). Fractal image compression based on intrablock variance distribution and vector quantization. *Optical Engineering* **41**(11):2824–2830.
- [23] H. Choi, R. G. Baraniuk (2001). Multiscale image segmentation using wavelet-domain hidden markov models. *IEEE Transactions on Image Processing* **10**(9):1309 – 1321.
- [24] R. R. Coifman, M. V. Wickerhauser (1992). Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory* **38**(2):713–718.
- [25] R. M. Corless et al. (1996). On the lambert w function. *Advances in Computational Mathematics* **5**:329–359.
- [26] Z. Cvetkovic, M. Vetterli (1998). Tight weyl-heisenberg frames in $l^2(z)$ **46**(5):1256–1259.
- [27] I. Daubechies (1988). Time-frequency localization operators: a geometric phase space approach. *IEEE Transactions on Information Theory* **34**(4):605–612.
- [28] I. Daubechies (1992). *Ten lectures on wavelets*. CBMS-NSF Regional Conference Series in Applied Mathematics 61. SIAM.
- [29] G. M. Davis, S. G. Mallat, M. Avellaneda (1997). Adaptive greedy approximations. *Journal of Constructive Approximation* **13**(1):57–98.
- [30] G. M. Davis, S. G. Mallat, Z. Zhang (1994). Adaptive time-frequency decompositions. *SPIE Journal of Optical Engineering* **33**(7):2183–2191.
- [31] L. Davis (ed.) (1991). *Handbook of Genetic Algorithms*. Van Nostrand.
- [32] S. R. Deans (1993). *The Radon transform and some of its applications*. Robert E. Krieger Publishing Co. Inc., Malabar, FL. Revised reprint of the 1983 original.
- [33] L. Demaret, H. Führ, F. Friedrich (2005). A quick guide to wedgelets. <http://www.wedgelet.de>.
- [34] R. A. DeVore (1998). Nonlinear approximation. *Acta Numerica* **7**:51–150.

-
- [35] R. A. DeVore, V. N. Temlyakov (1996). Some remarks on greedy algorithms. *Advances in Computational Mathematics* **5**(2-3):173–187.
- [36] S. J. Dilworth, D. Kutzarova, V. N. Temlyakov (2002). Convergence of some greedy algorithms in banach spaces. *Journal of Fourier Analysis and Applications* **8**(5):489 – 506.
- [37] O. Divorra Escoda (2005). *Toward Sparse and Geometry Adapted Video Approximations*. Ph.D. thesis, École Polytechnique Fédérale de Lausanne.
- [38] O. Divorra Escoda, L. Granai, P. Vandergheynst (2004). *On the Use of A Priori Information for Sparse Signal Approximations*. Tech. Rep. 23/2004, LTS2-ITS-EPFL.
- [39] O. Divorra Escoda, L. Granai, P. Vandergheynst (2004). *On the Use of A Priori Information for Sparse Signal Representations*. Tech. Rep. 18/2004, LTS2-ITS-EPFL.
- [40] O. Divorra Escoda, P. Vandergheynst (2004). A bayesian approach to video expansions on parametric over-complete 2-d dictionaries. In *International Workshop on Multimedia Signal Processing 2004 (MMSP 2004)*.
- [41] M. N. Do (2004). Contourlets: Construction and properties. IHP Summer School 2004 Slide support.
- [42] M. N. Do, P. L. Dragotti, R. Shukla, M. Vetterli (2001). On the compression of two dimensional piecewise smooth functions. In *Proceedings of the IEEE International Conference on Image Processing 2001 (ICIP '01)*.
- [43] M. N. Do, M. Vetterli (2001). Contourlets. In J. Stoecker, G. V. Welland (eds.), *Beyond Wavelets*, pp. 1–27, Academic Press, Inc.
- [44] M. N. Do, M. Vetterli (2002). Contourlets: A directional multiresolution image representation. In *Proceedings of the IEEE International Conference on Image Processing 2002 (ICIP '02)*, vol. 1, pp. 357–360, Rochester.
- [45] M. N. Do, M. Vetterli (To appear). The contourlet transform: An efficient directional multiresolution image representation. *IEEE Transactions on Image Processing* .
- [46] D. L. Donoho (1999). Wedgelets: Nearly-minimax estimation of edges. *Annals of Statistics* **27**(3):859–897.
- [47] D. L. Donoho, M. Elad (2003). Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization. *Proceedings of the National Academy of Sciences USA* **100**(5):2197–2202.
- [48] D. L. Donoho, M. Elad, V. Temlyakov (2004). *Stable Recovery of Sparse overcomplete Representations in the Presence of Noise*. Tech. rep., Industrial Mathematics Institute, University of South Carolina.
- [49] D. L. Donoho, X. Huo (2001). Uncertainty principles and ideal atomic decompositions. *IEEE Transactions on Information Theory* **47**(7):2845–2862.
- [50] P. L. Dragotti, M. Vetterli (2001). Footprints and edgeprints for image denoising and compression. In *Proc. International Conference on Image Processing 2001 (ICIP'01)*, vol. 2, pp. 237–240.

-
- [51] P. L. Dragotti, M. Vetterli (2003). Wavelet footprints: Theory, algorithms, and applications. *IEEE Transactions on Signal Processing* **51**(5):1306–1323.
- [52] R. Duffin, A. Schaffer (1952). A class of nonharmonic Fourier series. *Transactions of the American Mathematical Society* **72**(2):341–366.
- [53] P. J. Durka, D. Ircha, K. J. Blinowska (2001). Stochastic time-frequency dictionaries for matching pursuit. *IEEE Transactions on Signal Processing* **49**(3):507–510.
- [54] C. Duttweiler, D.L.and Chamzas (1995). Probability estimation in arithmetic and adaptive-huffman entropy coders **4**(3):237–246.
- [55] M. Elad, A. M. Bruckstein (2001). On sparse signal representations. In *Proceedings of the IEEE International Conference on Image Processing 2001 (ICIP '01)*, vol. 1, pp. 3–6, IEEE.
- [56] M. Elad, A. M. Bruckstein (2002). A generalized uncertainty principle and sparse representations in pairs of bases. *IEEE Transactions on Information Theory* **48**(9):2558–2567.
- [57] R. Eslami, H. Radha (2005). Wavelet-based contourlet packet image coding. In *2005 Conference on Information Sciences and Systems*, The Johns Hopkins University.
- [58] H. Everett III (1963). Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research* **11**:399–411.
- [59] M. D. Fairchild (1998). *Color Appearance Models*. Addison–Wesley Publishing Co.
- [60] A. Feuer, A. Nemirovski (2003). On sparse representation in pairs of bases. *IEEE Transactions on Information Theory* **49**(6):1579 – 1581.
- [61] R. M. Figueras i Ventura (2000). *Image Coding with Matching Pursuit*. Master’s thesis, Escola Tècnica Superior d’Enginyeria de Telecomunicació de Barcelona (ETSETB), Universitat Politècnica de Catalunya (UPC) and École Polytechnique Fédérale de Lausanne.
- [62] R. M. Figueras i Ventura, O. Divorra Escoda, P. Vandergheynst (2004). *A Matching Pursuit Full Search Algorithm for Image Approximations*. Tech. Rep. TR-ITS-31/2004, ITS-EPFL.
- [63] R. M. Figueras i Ventura, L. Granai, P. Vandergheynst (2002). *A generalized Rate-Distortion limit for edge representation*. Tech. Rep. 07.02, ITS-EPFL.
- [64] R. M. Figueras i Ventura, L. Granai, P. Vandergheynst (2002). R-D analysis of adaptive edge representations. In *IEEE Workshop on Multimedia Signal Processing 2002 (MMSP'02)*, pp. 130–133.
- [65] R. M. Figueras i Ventura, P. Vandergheynst (2001). *Matching Pursuit with Genetic Algorithms*. Tech. rep., F Group, LTS, EPFL.
- [66] R. M. Figueras i Ventura, P. Vandergheynst (2002). Evolutive multiresolution matching pursuit and its relation with the human visual system. In *Proceedings of the European Signal Processing Conference 2002 (EUSIPCO 2002)*, vol. 2, pp. 395–398, Toulouse, France.
- [67] R. A. Finkel, J. L. Bentley (1974). Quad trees, a data structure for retrieval on composite keys. *Acta Informatica* **4**:1–9.
- [68] A. G. Flesia et al. (2001). *Beyond Wavelets*, chap. Digital Implementation of Ridgelet Packets, pp. 1–34. Academic Press Inc.

-
- [69] W. T. Freeman, E. H. Adelson (1991). The design and use of steerable filters. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **13**(9):891–906.
- [70] J. H. Friedman, W. Stuetzle (1981). Projection pursuit regression. *Journal of the American Statistical Association* **76**:817–823.
- [71] J. H. Friedman, J. W. Tukey (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers* **23**(9):881–890.
- [72] P. Frossard (2000). *Robust and Multiresolution Video Delivery: From H.26x to Matching Pursuit Based Technologies*. Ph.D. thesis, EPFL.
- [73] P. Frossard, P. Vandergheynst (2001). A posteriori quantized matching pursuit. In *Proceedings of the IEEE Data Compression Conference, Snowbird*.
- [74] P. Frossard, P. Vandergheynst (2001). Redundancy in non-orthogonal transforms. In *Proceedings of the IEEE International Symposium on Information Theory 2001 (ISIT2001)*, p. 196, Washington D.C.
- [75] P. Frossard, P. Vandergheynst, R. M. Figueras i Ventura, M. Kunt (2004). A posteriori quantization of progressive matching pursuit streams. *IEEE Transactions of Signal Processing* **52**(2):525–535.
- [76] J.-J. Fuchs (2004). On sparse representations in arbitrary redundant bases. *IEEE Transactions on Information Theory* **50**(6):1341 – 1344.
- [77] J.-J. Fuchs (2004). *Recovery of exact sparse representations in the presence of bounded noise*. Tech. Rep. 1618, IRISA.
- [78] M. Gharavi-Alkhansari, T. S. Huang (1998). A fast orthogonal matching pursuit algorithm. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing 1998 (ICASSP 1998)*, vol. 3, pp. 1389–1392.
- [79] A. C. Gilbert, S. Muthukrishnan, M. J. Strauss (2003). Approximation of functions over redundant dictionaries using coherence. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 243 – 252, Society for Industrial and Applied Mathematics.
- [80] I. F. Gorodnitsky, B. D. Rao (1997). Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm. *IEEE Transactions on Signal Processing* **45**(3):600–616.
- [81] V. K. Goyal, M. Vetterli, N. T. Thao (1995). Quantization of overcomplete expansions. In *IEEE Proceedings of the Data Compression Conference 1995 (DCC'95)*, pp. 13 – 22, Snowbird, UT.
- [82] V. K. Goyal, M. Vetterli, N. T. Thao (1998). Quantized overcomplete expansions in \mathbb{R}^N : analysis, synthesis, and algorithms. *IEEE Transactions on Information Theory* **44**(1):16–31.
- [83] I. S. Gradshteyn, I. M. Ryzhik (1994). *Table of Integrals, Series, and Products*. Academic Press, Inc., fifth edn.
- [84] R. Gribonval, R. M. Figueras i Ventura, P. Vandergheynst (2004). *A simple test to check the optimality of sparse signal approximations*. Tech. Rep. 1661, IRISA.

-
- [85] R. Gribonval, R. M. Figueras i Ventura, P. Vandergheynst (2005). A simple test to check the optimality of sparse signal approximations. In *Proceedings of Acoustics, Speech, and Signal Processing 2005 (ICASSP '05)*, vol. 5, pp. 717 – 720.
- [86] R. Gribonval, M. Nielsen (2003). Approximation with highly redundant dictionaries. In *Proceedings of SPIE 2003 Wavelets: Applications in Signal and Image Processing*, vol. 5207, pp. 216–227, SPIE.
- [87] 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.
- [88] R. Gribonval, M. Nielsen (2003). Sparse decompositions in "incoherent" dictionaries. In *Proceedings of the IEEE International Conference on Image Processing 2003 (ICIP '03)*, vol. 1, pp. 33–36, IEEE.
- [89] R. Gribonval, M. Nielsen (2003). Sparse representations in unions of bases. *IEEE Transactions on Information Theory* **49**(12):3320–3325.
- [90] R. Gribonval, M. Nielsen (2004). Nonlinear approximation with dictionaries i. direct estimates. *Journal of Fourier Analysis and Applications* **10**(1):51–71.
- [91] R. Gribonval, P. Vandergheynst (2004). *On the exponential convergence of Matching Pursuit in quasi-incoherent dictionaries*. Tech. rep., IRISA.
- [92] A. C. Guyton, J. E. Hall (1996). *Textbook of Medical Physiology*. W.B. Saunders Company, United States of America, ninth edn.
- [93] P. J. B. Hancock, R. J. Baddeley, L. S. Smith (1992). The principal components of natural images. *Network* **3**(1):61–70. Printed in the U.K.
- [94] Y. Hel-Or, P. C. Teo (1996). *A Common Framework for Steerability, Motion Estimation and Invariant Feature Detection*. Tech. Rep. CS-TN-96-28, Stanford University.
- [95] Y. Hel-Or, P. C. Teo (1998). Canonical decomposition of steerable functions. *Journal of Mathematical Imaging and Vision* **9**(1).
- [96] D. H. Hubel, T. N. Wiesel (1959). Receptive fields of single neurons in the cat's striate cortex. *Journal of Physiology* **148**:574–491. London.
- [97] A. K. Jain, F. Farrokhnia (1991). Unsupervised texture segmentation using gabor filters. *Pattern Recognition* **24**(12):1167–1186.
- [98] T. W. James et al. (2000). The effects of visual object priming on brain activation before and after recognition. *Current Biology* **10**(17):1017–1024.
- [99] L. K. Jones (1987). On a conjecture of Huber concerning the convergence of projection pursuit regression. *The Annals of Statistics* **15**(2):880–882.
- [100] P. Jost, P. Vandergheynst, P. Frossard (2005). Tree-based pursuit: Algorithm and properties. *IEEE Transactions on Signal Processing* **submitted**.

-
- [101] R. Koenen (2002). *Overview of the MPEG-4 Standard*. International Organisation for Standardisation Organisation Internationale de Normalisation ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Audio. <http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm>.
- [102] H. Kolb (2003). How the retina works. *American Scientist* **91**(1):28–35.
- [103] H. Kolb, E. Fernandez, N. Ralph (2002). Webvision: The organization of the retina and visual system. webpage: <http://webvision.med.utah.edu/>.
- [104] S. V. Konyagin, V. N. Temlyakov (2002). *Convergence of greedy approximation I. General systems*. Tech. Rep. 02:08, Industrial Mathematics Institute, University of South Carolina.
- [105] S. V. Konyagin, V. N. Temlyakov (2002). *Convergence of greedy approximation II. The triangular system*. Tech. Rep. 02:09, Industrial Mathematics Institute, University of South Carolina.
- [106] S. V. Konyagin, V. N. Temlyakov (2004). *Convergence of greedy algorithms for the trigonometric system*. Tech. Rep. 2004:14, Industrial Mathematics Institute, University of South Carolina.
- [107] A. P. Korostelev, A. B. Tsybakov (1993). *Minimax Theory of Image Reconstruction*, vol. 82 of *Lecture Notes in Statistics*. Springer-Verlag.
- [108] K. Kreutz-Delgado et al. (2003). Dictionary learning algorithms for sparse representation. *Neural Computation* **15**(2):349–396. Letter.
- [109] E. Le Pennec, S. Mallat (2001). Bandelet representation for image compression. In *Proceedings of the IEEE International Conference on Image Processing 2001 (ICIP '01)*, vol. 1, pp. 12–15.
- [110] E. Le Pennec, S. Mallat (2005). Sparse geometrical image approximation with bandelets. *IEEE Transaction on Image Processing* **(to appear)**.
- [111] D. Leviatan, V. N. Temlyakov (2003). *Simultaneous greedy approximation in Banach spaces*. Tech. rep., Industrial Mathematics Institute, University of South Carolina.
- [112] D. Leviatan, V. N. Temlyakov (2005). Simultaneous approximation by greedy algorithms. *Advances in Computational Mathematics* **(to appear)**.
- [113] M. W. Levine, J. M. Shefner (1991). *Fundamentals of sensation and perception*, p. 671. Pacific Grove, CA: Brooks/Cole, 2 edn.
- [114] T. Lindeberg (1998). Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision* **30**(2):117–156.
- [115] Y. Lu, M. N. Do (2003). CRISP-contourlets: a critically sampled directional multiresolution image representation. In *Proceedings of SPIE conference on Wavelet Applications in Signal and Image Processing*.
- [116] A. Lutoborski, V. N. Temlyakov (2003). Vector greedy algorithms. *Journal of Complexity* **19**(4):458–473.
- [117] D. M. Malioutov, M. Cetin, A. S. Willsky (2004). Optimal sparse representations in general overcomplete bases. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing 2004 (ICASSP '04)*, vol. 2, pp. 793–796.

-
- [118] S. Mallat (1998). *A wavelet tour of signal processing*. Academic Press.
- [119] S. Mallat, Z. Zhang. (1993). Matching pursuit with time-frequency dictionaries. *IEEE Transactions on Signal Processing* **41**(12):3397–3415.
- [120] R. Manduchi, P. Perona, D. Shy (1997). *Efficient Implementation of Deformable Filter Banks*. Tech. Rep. CNS-TR-97-04, California Institute of Technology, <http://www.vision.caltech.edu/manduchi/deformable.ps.Z>.
- [121] B. Manjunath, W. Ma (1996). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(8):837–842.
- [122] D. H. Marimont, Y. Rubner (1998). A probabilistic framework for edge detection and scale selection. In *Computer Vision, 1998. Sixth International Conference on*, IEEE, Bombay, India.
- [123] D. Marr (1976). Early processing of visual information. *Philosophical Transactions of the Royal Society of London. A. : Biological Sciences* **275**(942):483–524.
- [124] D. Marr (1982). *Vision*. W. H. Freeman and Company.
- [125] D. Marr, E. Hildreth (1980). Theory of edge detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences* **207**(1167):187–217.
- [126] c. Mendels, Fran (2004). *Matching Pursuit-based shape description and recognition in active contours*. Ph.D. thesis, École Polytechnique Fédérale de Lausanne.
- [127] Y. Meyer (2002). Oscillating patterns in image processing and in some nonlinear evolution equations. In *AMS*, vol. 22 of *University Lecture Series*.
- [128] A. J. Miller (2002). *Subset Selection in Regression*. Chapman & Hall/CRC, London, 2 edn.
- [129] G. Monaci, O. Divorra Escoda, P. Vandergheynst (2005). Analysis of multimodal signals using redundant representations. In *Proceedings of the IEEE International Conference on Image Processing 2005 (ICIP'05)*.
- [130] J. F. Murray, K. Kreutz-Delgado (2001). An improved FOCUSS-based learning algorithm for solving sparse linear inverse problems. In *Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on*, vol. 1, pp. 347 – 351, IEEE.
- [131] M. Nadenau (2000). *Integration of Human Color Vision Models into High Quality Image Compression*. Ph.D. thesis, École Polytechnique Fédérale de Lausanne, Lausanne.
- [132] P. B. Nair, A. Choudhury, A. J. Keane (2003). Some greedy learning algorithms for sparse regression and classification with mercer kernels. *The Journal of Machine Learning Research* **3**(SPECIAL ISSUE: Special issue on ICML):781 – 801.
- [133] B. S. Nalwa, T. O. Binford (1986). On detecting edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(6):699 – 714.
- [134] B. K. Natarajan (1995). Sparse approximate solutions to linear systems. *SIAM Journal on Computing* **24**(2):227–234.
- [135] R. Neff, A. Zakhor (1997). Very low bit-rate video coding based on matching pursuits. *IEEE Transactions on Circuits and Systems for Video Technology* **7**(1):158–171.

-
- [136] R. Neff, A. Zakhor (1999). Adaptive modulus quantizer design for matching pursuit video coding. In *Proceedings of the IEEE International Conference on Image Processing 1999 (ICIP 99)*, vol. 2, pp. 81–85.
- [137] R. Neff, A. Zakhor (2000). Modulus quantization for matching-pursuit video coding. *IEEE Transactions on Circuits and Systems for Video Technology* **10**(6):895–912.
- [138] B. A. Olshausen, D. J. Field (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **13**(381(6583)):607 – 609.
- [139] B. A. Olshausen, D. J. Field (1997). Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research* **37**(23):3311–3325.
- [140] B. A. Olshausen, D. J. Field (2004). *23 Problems in Systems Neuroscience*, chap. What is the Other 85 Oxford University Press.
- [141] A. Ortega, K. Ramchandran (1998). Rate-distortion methods for image and video compression. *IEEE Signal Processing Magazine* **15**(6):23–50.
- [142] J. Ostermann, E. S. Jang, J.-S. Shin, T. Chen (1997). Coding of arbitrarily shaped video object in mpeg-4. In *Proceedings of the IEEE International Conference on Image Processing 1997 (ICIP '97)*, vol. 1, pp. 496–499.
- [143] Y. C. Pati, R. Rezaifar, P. S. Krishnaprasad (1993). Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Conference Record of the 27th Annual Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 40–44, Pacific Grove, CA, USA.
- [144] L. Peotta, L. Granai, P. Vandergheynst (2005). Image compression using an edge adapted redundant dictionary and wavelets. *Signal Processing to appear*.
- [145] P. Perona (1992). Steerable-scalable kernels for edge detection and junction analysis. *IVC* **10**:663–672.
- [146] S. Pomerants, James R. amd Jewell (2005). Psychology of perception (images). Webpage: <http://www.owlnet.rice.edu/psyc351/>.
- [147] V. Popovici (2004). *Kernel-based classifiers with applications to face detection*. Ph.D. thesis, École polytechnique Fédérale de Lausanne.
- [148] K. Prachumrak, A. Hiramatsu, T. Fuchida, H. Nakamura (2003). Lossless fractal image coding. In *Video/Image Processing and Multimedia Communications, 2003. 4th EURASIP Conference focused on*, vol. 1, pp. 95 – 100, EURASIP.
- [149] W. K. Pratt (2001). *Digital Image Processing*. John Wiley & Sons, Inc.
- [150] A. Puri, A. Eleftheriadis (1998). MPEG-4: an object-based multimedia coding standard supporting mobile applications. *Mobile Networks and Applications* **3**(1):5 – 32.
- [151] K. Ramchandran, M. Vetterli (1993). Best wavelet packet bases in a rate-distortion sense. *IEEE Transactions on Image Processing* **2**(4):160–175.
- [152] L. Rebollo-Neira, D. Lowe (2002). Optimized orthogonal matching pursuit approach. *IEEE Signal Processing Letters* **9**(4):137–140.

-
- [153] I. Richardson (2002). *Video Codec Design: Developing Image and Video Compression Systems*. Wiley, New York.
- [154] H. Romberg, Justin K. and Choi, R. G. Baraniuk (2001). Bayesian tree-structured image modeling using wavelet-domain hidden markov models. *IEEE Transactions on Image Processing* **10**(7):1056–1068.
- [155] J. K. Romberg, M. B. Wakin, R. G. Baraniuk (2003). Approximation and compression of piecewise smooth images using a wavelet/wedgelet geometric model. vol. 1, pp. 49–52.
- [156] W. Rudin (1976). *Principles of Mathematical Analysis*. McGraw-Hill.
- [157] W. Rudin (1987). *Real and Complex Analysis*. Mc Graw Hill.
- [158] A. Said, W. Pearlman (1993). Reversible image compression via multiresolution representation and predictive coding. In *Proc. SPIE-VCIP 1993*, pp. 664–674. [Http://www.cipr.rpi.edu/research/SPIHT/](http://www.cipr.rpi.edu/research/SPIHT/).
- [159] A. Said, W. A. Pearlman (1996). A new, fast, and efficient image codec based on set partitioning in hierarchical trees **6**(3):243–250.
- [160] H. Samet (1990). *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS*. Addison-Wesley.
- [161] H. Samet (1990). *The Design and Analysis of Spatial Data Structures*. Addison-Wesley.
- [162] S. Sardy, A. G. Bruce, P. Tseng (2000). Block coordinate relaxation methods for nonparametric wavelet denoising. *Journal of Computational and Graphical Statistics* **9**(2):361–379.
- [163] D. Saupe, R. Hamzaoui, H. Hartenstein (1997). *Fractal Image Compression - An Introductory Overview*. Tech. Rep. 92, Institut für Informatik, Universität Freiburg, Germany. <http://www.informatik.uni-freiburg.de/tr/1997/>.
- [164] K. Sayood (2000). *Introduction to Data Compression*. Academic Press, 2nd edn.
- [165] E. Schmidt (1906-1907). Zur Theorie der linearen und nichtlinearen Integralgleichungen. *Mathematische Annalen* **63**:433–476.
- [166] G. M. Schuster, A. K. Katsaggelos (1998). An optimal quadtree-based motion estimation and motion-compensated interpolation scheme for video compression. *IEEE Transactions on Image Processing* **7**(11):1505–1523.
- [167] G. M. Schuster, G. Melnikov, A. K. Katsaggerlos (1998). Operational optimal vertex-based shape coding. *IEEE Signal Processing Magazine* pp. 91–108.
- [168] O. Schwartz, E. P. Simoncelli (2001). Natural signal statistics and sensory gain control. *Nature Neuroscience* **4**(8):819–825.
- [169] Serendip (1994). Seeing more than your eye does. Web page. <http://serendip.brynmawr.edu/bb/blindspot1.html>.
- [170] D. Shanno, R. J. Vanderbei (2000). Interior-point methods for nonconvex nonlinear programming: Orderings and higher-order methods. *Mathematical Programming* **87**(2):303–316.
- [171] Y. Shoham, A. Gersho (1988). Efficient bit allocation for an arbitrary set of quantizers. *IEEE Transactions on Acoustics, Speech and Signal Processing* **36**(9):1445–1453.

-
- [172] R. Shukla, P. L. Dragotti, M. N. Do, M. Vetterli (2005). Rate-distortion optimized tree-structured compression algorithms for piecewise polynomial images. *IEEE Transactions on Image Processing* **14**(3):343–359.
- [173] E. P. Simoncelli (1997). Statistical models for images: compression, restoration and synthesis. In *Signals, Systems & Computers, 1997. Conference Record of the Thirty-First Asilomar Conference on,*, vol. 1, pp. 673–678, IEEE.
- [174] E. P. Simoncelli (2003). Vision and the statistics of the visual environment. *Current Opinion in Neurobiology* **13**(2):144–149.
- [175] E. P. Simoncelli, B. A. Olshausen (2001). Natural image statistics and neural representation. In *Annu Rev Neurosci. 2001*, vol. 24, pp. 1193–1216.
- [176] E. P. Simoncelli, J. Portilla (2000). A parametric texture based model based on joint statistics of complex wavelet coefficients. *Int'l Journal of Computer Vision* pp. 1–38.
- [177] A. Srivastava, A. B. Lee, E. P. Simoncelli, S.-C. Zhu (2003). On advances in statistical modeling of natural images. *Journal of Mathematical Imaging and Vision* **18**(1):17–33. Kluwer Academic Publishers.
- [178] J.-L. Starck, E. J. Candès, D. L. Donoho (2002). The curvelet transform for image denoising. *IEEE Transactions on Image Processing* **11**(6):670–684.
- [179] G. J. Sullivan (1996). Efficient scalar quantization of exponential and laplacian random variables. *IEEE Transactions on Information Theory* **42**(5):1365–1374.
- [180] D. Taubman, M. Marcellin (2001). *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, Boston.
- [181] D. Taubman, A. Zakhor (1994). Multirate 3-D subband coding of video. *IEEE Transactions on Image Processing* **3**(5):572–588.
- [182] V. N. Temlyakov (2000). Weak greedy algorithms. *Advances in Computational Mathematics* **12**(2-3):213 – 227.
- [183] V. N. Temlyakov (2001). Greedy algorithms in Banach spaces. *Advances in Computational Mathematics* **14**(3):277–292.
- [184] V. N. Temlyakov (2002). A criterion for convergence of weak greedy algorithms. *Advances in Computational Mathematics* **17**(2):269–280.
- [185] V. N. Temlyakov (2003). Nonlinear methods of approximation. *Foundations of Computational Mathematics* **3**(1):33 – 107.
- [186] V. N. Temlyakov (2004). *On greedy algorithms with restricted depth search*. Tech. Rep. 2004:27, Industrial Mathematics Institute, University of South Carolina.
- [187] M. E. Tipping (2001). Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research* **1**(3):211–244.
- [188] J. A. Tropp (2004). Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory* **50**(10):2231–2242.

-
- [189] J. A. Tropp (2004). *Just Relax: Convex Programming Methods for Subset Selection and Sparse Approximation*. Tech. Rep. 04-04, Institute for Computational Engineering and Sciences (ICES), The University of Texas at Austin.
- [190] J. A. Tropp (2004). *Topics in Sparse Approximation*. Ph.D. thesis, The University of Texas at Austin.
- [191] J. A. Tropp, A. C. Gilbert, S. Muthukrishnan, M. J. Strauss (2003). Improved sparse approximation over quasi-incoherent dictionaries. In *Proceedings of the IEEE International Conference on Image Processing 2003 (ICIP '03)*, vol. 1, pp. 37–40.
- [192] S. Ullman (1997). *High-level Vision*. MIT Press, second edn.
- [193] A. van de Walle (1997). Merging fractal image compression and wavelet transform methods. *Fractals* **5**(Supplementary Issue):3–15.
- [194] R. J. Vanderbei (2001). *Linear Programming: Foundations and Extensions*, vol. 37 of *International Series in Operations Research & Management Science*. Springer, 2 edn.
- [195] P. Vandergheynst, P. Frossard (2001). Efficient image representation by anisotropic refinement in matching pursuit. In *Proc. ICASSP 2001*, vol. 3, pp. 1757–1760.
- [196] L. A. Vese, S. J. Osher (2003). Modeling textures with total variation minimization and oscillating patterns in image processing. *Journal of Scientific Computing* **19**(1-3):553–572.
- [197] M. Vetterli (2001). Wavelets, approximation and compression. *IEEE Signal Processing Magazine* **18**(5):59 – 73.
- [198] L. F. Villemoes (1996). Nonlinear approximation with walsh atoms. In A. Le Méhauté, C. Rabut, L. L. Schumaker (eds.), *Surface fitting and multiresolution methods*, pp. 329–336, Vanderbilt University Press, Chamonix, France.
- [199] L. F. Villemoes (1997). Best approximation with walsh atoms. *Constructive Approximation* **13**(3):329–355.
- [200] L. Vincent, P. Soille (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(6):583 – 598.
- [201] M. J. Wainwright, O. Schwartz, E. P. Simoncelli (2001). *Statistical Theories of the Brain*, chap. Natural Images Statistics and Divisive Normalization: Modeling Nonlinearities and Adaption in Cortical Neurons. MIT Press.
- [202] M. J. Wainwright, E. P. Simoncelli (2000). Scale mixtures of gaussians and the statistics of natural images. In T. K. Leen, K.-R. Müller (eds.), *Advances in Neural Information Processing Systems*, vol. 12, pp. 855–861, MIT Press.
- [203] M. J. Wainwright, E. P. Simoncelli, A. S. Willsky (2000). Random cascades on wavelet trees and their use in analyzing and modeling natural images. In *Proceedings of the 45th Annual Meeting of the SPIE, SPIE*.
- [204] M. Wakin, J. Romberg, H. Choi, R. Baraniuk (2002). Image compression using an efficient edge cartoon + texture model. In *Data Compression Conference*, pp. 43–52.

-
- [205] M. Wakin, J. Romberg, H. Choi, R. Baraniuk (2002). Rate-distortion optimized image compression using wedgelets. In *IEEE International Conference on Image Processing 2002 (ICIP'02)*, vol. 3, pp. 237–240.
- [206] M. Wakin, J. Romberg, H. Choi, R. Baraniuk (2003). Geometric methods for wavelet-based image compression. In M. A. Unser, A. Aldroubi, A. F. Laine (eds.), *International Symposium on Optical Science and Technology*, vol. 5207, pp. 507–520, SPIE, San Diego, CA.
- [207] Z. Wang, A. C. Bovik (2001). Embedded foveation image coding. *IEEE Transactions on Image Processing* **10**(10):1397–1410.
- [208] S. Westen, R. Lagendijk, J. Biemond (1996). Optimization of jpeg color image coding using a human visual system model. In *Proceedings of the SPIE Conference on Human Vision and Electronic Imaging*, vol. 2657, pp. 370–381, San Jose, USA.
- [209] D. P. Wipf, B. D. Rao (2003). Bayesian learning for sparse signal reconstruction. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing 2003 (ICASSP '03)*, vol. 6, pp. 601–604, IEEE.
- [210] I. H. Witten, R. M. Neal, J. G. Cleary (1987). Arithmetic coding for data compression. *Communications of the ACM* **30**(6):520–540.
- [211] B. Wohlberg (2003). Noise sensitivity of sparse signal representations: Reconstruction error bounds for the inverse problem. *IEEE Transactions on Signal Processing* **51**(12):3053–3060.
- [212] J. W. Woods, G. Lilienfeld (2001). A resolution and frame-rate scalable subband/wavelet video coder **11**(9):1035–1044.
- [213] M. Zibulevsky, B. Pearlmutter (2001). Blind source separation by sparse decomposition in a signal dictionary. *Neural Computations* **13**(4):863–882.

Curriculum Vitae

Name: **Rosa M. Figueras i Ventura**
Citizenship: Catalan (Spanish)
Birthdate: January 29, 1977
Birthplace: Banyoles, Catalonia, Spain
Email: Rosa.Figueras@epfl.ch

Work experience

February 2000 – present: Research in signal and image processing, with special focus on image coding, École Polytechnique Fédérale de Lausanne, Switzerland

- Research in signal and image processing, with special focus on image coding
 - Study of the approximation properties of different sets of functions (dictionaries)
 - Implementation of an image coder through non-linear approximations, with special properties allowing on the fly geometrical image transformations
- Mathematical study of the approximation properties of geometric dictionaries
 - Implementation of a software to verify these mathematical properties
- Supervision of student projects and diploma
- Supervision and preparation of laboratory courses

April 2004: One month internship at IRISA, Rennes (France).

- Mathematical study of the properties of sparse approximations, in collaboration with R. Gribonval

August 2002: Collaboration with the organization of IEEE International Conference on Multimedia and Expo (ICME).

- Technical and logistical support.

June – September 1999: Praktikum at Robert Bosch GmbH (Hildesheim, Germany).

- Development of Virtual Home Environment Applications (multi terminal applications that can be seen exactly the same through any media)
- Internet Security Study

Education

February 2001 – June 2005: Ph.D. in Image Coding and Representation, Signal Processing Institute (ITS), École Polytechnique Fédérale de Lausanne(EPFL), Switzerland.

February 2000 – August 2000: Erasmus exchange student. Master thesis at the Signal Processing Laboratory, (EPFL), Lausanne, Switzerland, on new techniques of image coding for very low bit-rate applications, with honours.

September 1995 – September 2000: Telecommunication Engineer at Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona (ETSETB), Technical University of Catalonia (UPC), Barcelona, Catalonia, Spain. Speciality: Communications.

Post-graduate Education

September 2004: Attendance to the Seminar for Applied Mathematics IHP Summer School, Zurich, Switzerland. Duration: one week.

June 2004: Attendance to the course PDEs in image processing , by G. Sapiro, UPC, Barcelona, Catalonia, Spain. Duration: one week

December 2003: HASSIP Minischool on Non-linear and Adaptive Approximation, Marseille, France. Duration: one week.

April 2001: Postgraduate course on Image and Signal Processing (EPFL), Lausanne, Switzerland. Duration: two weeks.

Languages

Catalan → mother tongue
Spanish → spoken as if mother tongue
English → fluent oral and written
French → fluent oral and written
German → medium knowledge
Italian → medium knowledge

Publications

Journal Papers

- Rémi Gribonval, Rosa M. Figueras i Ventura and Pierre Vandergheynst, **A simple test to check the optimality of sparse signal approximations**. Accepted in EURASIP Signal Processing Journal, Special Issue on Sparse Approximations in Signal and Image Processing
- Rosa M. Figueras i Ventura, Pierre Vandergheynst and Pascal Frossard, **Low rate and scalable image coding with redundant representations**. Accepted in IEEE Transactions on Image Processing.
- Pascal Frossard, Pierre Vandergheynst, Rosa M. Figueras i Ventura and Murat Kunt, **A Posteriori Quantization of Progressive Matching Pursuit Streams**, IEEE Transactions on Signal Processing, Vol. 52, No 2, pp. 525-535, February 2004

Conference Papers

- Rémi Gribonval, Rosa M. Figueras i Ventura and Pierre Vandergheynst, **A simple test to check the optimality of sparse signal approximations**. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing 2005 (ICASSP '05), Volume 5, pp. 717-720, March 2005.
- Rosa M. Figueras i Ventura, Pierre Vandergheynst, Pascal Frossard and Andrea Cavallaro, **Color image scalable coding with Matching Pursuit**. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing 2004 (ICASSP '04), Volume III, pp. 53-56, Montreal, May 2004.
- Pascal Frossard, Pierre Vandergheynst and Rosa M. Figueras i Ventura, **High flexibility scalable image coding**. In Proceedings of VCIP 2003, Lugano, July 2003.
- Rosa M. Figueras i Ventura, Lorenzo Granai and Pierre Vandergheynst, **R-D Analysis of Adaptive Edge Representations**. 2002 IEEE workshot on Multimedia Signal Processing, pp. 130-133, December 2002.
- Rosa M. Figueras i Ventura, Pierre Vandergheynst and Pascal Frossard, **Evolutionary Multiresolution Matching Pursuit and its relations with the Human Visual System**. In Proceedings of EUSIPCO 2002, Volume II pp. 395-398, Toulouse, France, September 2002.

- Touradj Ebrahimi, Yousri Abdeljaoued, Rosa M. Figueras i Ventura and Òscar Divorra Escoda, **MPEG-7 CAMERA**. In Proceedings of IEEE ICIP, Thessaloniki, Greece, October 2001.
- Òscar. Divorra Escoda, Rosa M. Figueras i Ventura, Eric Debes and Touradj Ebrahimi, **Influence of a large image watermarking scheme parallelization on possible attacks**. In Proceedings of SPIE 46th Annual Conference, San Diego, July 2001.

Technical Reports

- Rosa M. Figueras i Ventura, Òscar Divorra Escoda and Pierre Vandergheynst, **A Matching Pursuit Full Search Algorithm for Image Approximations**, ITS Technical Report TR-ITS-31/2004, December 2004
- Rémi Gribonval, Rosa M. Figueras i Ventura and Pierre Vandergheynst, **A simple test to check the optimality of a sparse signal approximation**, IRISA Technical Report N. 1661, November 2004
- Rosa M. Figueras i Ventura, Pierre Vandergheynst, Pascal Frossard and Andrea Cavallaro, **Color Image Scalable Coding with Matching Pursuit**. ITS Technical Report TR-ITS-05.03, October 2003
- Rosa M. Figueras i Ventura, Pierre Vandergheynst and Pascal Frossard, **Low rate and scalable image coding with redundant representations**. ITS Technical Report TR-ITS-03.02, June 2003
- Rosa M. Figueras i Ventura, Lorenzo Granai and Pierre Vandergheynst, **A generalized Rate-Distortion limit for edge representation**. LTS Technical Report 02.07, May 2002
- Rosa M. Figueras i Ventura, Pierre Vandergheynst, and Pascal Frossard, **Evolutionary Multiresolution Matching Pursuit and its relations with the Human Visual System**. LTS Technical Report 01.04, December 2001
- Rosa M. Figueras i Ventura and Pierre Vandergheynst, **Matching Pursuit through Genetic Algorithms**. LTS Technical Report 01.02, July 2001

Master Thesis

- Rosa Maria Figueras i Ventura, **Image Coding with Matching Pursuit**, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, August 2000 and Technical University of Catalonia (UPC).