8

Image coding using redundant dictionaries

Pierre Vandergheynst, Pascal Frossard

Signal Processing Institute
Swiss Federal Institute of Technology, EPFL - Switzerland

Abstract

This chapter discusses the problem of coding images using very redundant libraries of waveforms, also referred to as dictionaries. We start with a discussion of the shortcomings of classical approaches based on orthonormal bases. More specifically, we show why these redundant dictionaries provide an interesting alternative for image representation. We then introduce a special dictionary of 2-D primitives called anisotropic refinement atoms that are well suited for representing edge-dominated images. Using a simple greedy algorithm, we design an image coder that performs very well at low bit rate. We finally discuss its performance and particular features such as geometric adaptativity and rate scalability.

8.1 Introduction

Image compression has been key in enabling what already seems to be two of the major success stories of the digital era: rich media experience over the internet and digital photography. What are the technologies laying behind such industry flagships and, more importantly, what are the future of these technologies are some of the central questions of this chapter.

We begin with a quick review of the state of the art. Then, identifying some weaknesses of the actual systems together with new requirements from applications, we depict how novel algorithms based on redundant libraries could lead to new breakthroughs.

8.1.1 A quick glance at digital image compression

Modern image compression algorithms, most notably JPEG, have been designed following the transform coding paradigm. Data is considered as the output symbols x_n of a random source X, which can have a complicated probability density function. In order to reduce redundancy between symbols, one seeks a new representation of the source by applying a suitable linear transformation T. The new symbols $Y = T \cdot X$ will then be quantized and entropy coded. Very often a scalar quantizer will be applied to the transform coefficients y_n. It is a standard result in data compression that, in order to maximize the performance a such a system, the transform T should be chosen so that it yields uncorrelated coefficients. In this regards, the optimal transform is thus the Karhunen-Loeve Transform (KLT). It is one of the beauty of transform coding that such a simple and complete analysis is possible. It also leads us to a few important comments about the whole methodology. First, the KLT is a data-dependent and complex transform. Using it in practice is at least difficult, usually impossible as it would require to send the basis that represents T to the decoder for each source. Very often, one seeks a linear transform that performs close to the KLT and this is one of the reasons why the DCT was chosen in JPEG. Second, the optimality of the transform coding principle (KLT plus scalar quantizer) can only be ensured for simple models (e.g., gaussian cyclostationary). In practice, for natural data, this kind of modelling is far from truth. Finally, the role of the transform in this chain is relegated to its role of providing uncorrelated coefficients for feeding the scalar quantizer. Nothing about the main structures of the signal and the suitability of the transform to catch them is ever used.

Based on these observations the research community started to consider other alternatives:

- Replacing scalar quantization by Vector Quantization (VQ), which can be seen as a way to overcome the limits of transform coding while also putting more emphasis on the content of the signal.
- Searching and studying new transforms, better suited to represent the content of the signal.
- Completely replacing transform coding by other techniques.

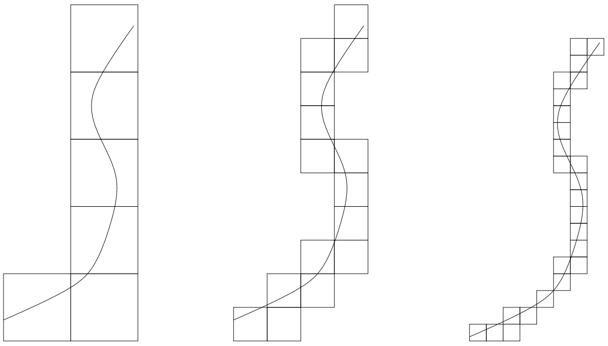
Out of the many interesting techniques that have emerged based on these interrogations, wavelet based techniques have had the largest impact.

Indeed, these last few years, image compression has been largely dominated by the use of wavelet based transform coding techniques. Many popular compression algorithms use wavelets at their core (SPIHT, EBCOT) and the overall success of this methodology resulted in the actual JPEG2000 standard for image compression [1]. As it was quickly realized, there is more to wavelets than their simple use as a decorrelating transform. On the conceptual point of view, we see three main reasons for their success: (i) fast algorithms based on filter banks or on the lifting scheme, (ii) nice mathematical properties, and (iii) smart adaptive coding of the coefficients.

Efficient algorithms are, of course, of paramount importance when putting a novel technique to practice, but the overall power of wavelets for image compression really lies in the second and third items. The mathematical properties of wavelets have been well studied in the fields of Computational Harmonic Analysis (CHA) and Non-Linear Approximation Theory. Generally, the central question that both theories try to answer (at least in connection with data compression) is: given a signal, how many wavelet coefficients do I need to represent it up to a given approximation error? There is a wealth of mathematical results that precisely relate the decay of the approximation error with the smoothness of the original signal, when N coefficients are used. Modelling a signal as a piecewise smooth function, it can be shown that wavelets offer the best rate of non-linear approximation. By this we mean that approximating functions that are locally Hölder α with discontinuities, by their N biggest wavelet coefficients, one obtains an approximation error in the order of $N^{-\alpha}$ and that this is an optimal result (see [2, 3] and references therein). The key to this result is that wavelet bases yield very sparse representations of such signals, mainly because their vanishing moments *kill* polynomial parts, while their multiresolution behavior allows to localize discontinuities with few non-negligible elements. Now, practically speaking, the real question should be formulated in terms of bits: how many bits do I need to represent my data up to a given distortion? The link between both questions is not really trivial: it has to take into account both quantization and coding strategies. But very efficient wavelet coding schemes exist, and many of them actually use the structure of non-negligible wavelet coefficients accross subbands.

8.1.2 Limits of current image representation methods

While the situation described above prevails in one dimension, it gets much more problematic for signals with two or more dimensions, mainly because of the importance of geometry. Indeed, an image can still be modeled as a piecewise smooth 2-D signal with singularities, but the latter are not point like anymore. Multi-dimensional singularities may be highly organized along embedded submanifolds and this is exactly what happens at image contours for example. Figure 8.1 shows that wavelets are inefficient at representing contours because they cannot deal with the geometrical regularity of the contours themselves. This is mainly due to the isotropic refinement implemented by wavelet basis: the dyadic scaling factor is applied in all directions, where clearly it should be fine along the direction of the local gradient and coarse in the orthogonal direction in order to efficiently *localize* the singularity in a sparse way. This is the reason why other types of signal representation, like redundant transforms, certainly represent the core of new breakthroughs in image coding, beyond the performance of orthogonal wavelet transforms.



(a) 6 coefficients (b) 15 coefficients (c) 25 coefficients

Figure 8.1: Inadequacy of isotropic refinement for representing contours in images. The number of wavelets intersecting the singularity is roughly doubled when the resolution increases.

8.2 Redundant expansions

8.2.1 Benefits of redundant transforms

In order to efficiently represent contours, beyond the performance of wavelet decompositions, anisotropy is clearly desirable in the coding scheme. Several authors have explored the rate-distortion characteristics of anisotropic systems for representing edge-dominated images [4, 5]. These preliminary studies show that for images that are smooth away from a smooth edge (typically a $\mathcal{C}^2$ rectifiable curve), a rate-distortion (R-D) behavior of the form

$$D(R) \asymp \frac{\log R}{R^2} \tag{8.1}$$

can be reached. Comparing this with the associated wavelet R-D behavior, i.e., $D(R) \asymp R^{-1}$, one clearly sees how the use of a geometry-adapted system of representation can boost coding expectations. It is important to realize here that it is really the anisotropic scaling of the basis functions that allows for such performances. Simply using an anisotropic basis with multiple orientations but a fixed isotropic scaling law would not provide such results (though it may improve visual quality for instance).

Candes and Donoho [6] have recently proposed a construction called *the curvelet transform* which aims at solving the lack of flexibility of wavelets in higher dimensions. Basically curvelets satisfy an anisotropic scaling law that is adapted to representing smooth curves in images. Curvelet tight frames have been shown to achieve a much better non-linear approximation rate than wavelets for images that are smooth away from a $\mathcal{C}^2$ edge. Very interesting results have been reported for statistical estimation and denoising [7] and efficient filter bank implementations have been designed [8]. On the coding side, curvelets satisfy the localization properties that lead to (8.1) and there is thus hope to find efficient compression schemes based on the curvelet transform, even though such results have not yet been reported.

8.2.2 Non-Linear Algorithms

A wealth of algorithms

Clearly, another way to tackle the problem of higher dimensional data representation would be to turn to non-linear algorithms. The interested reader searching a way through the literature might feel as if he/she had suddenly opened Pandora's box! Various algorithms exist, and they all differ in philosophy. Before moving to the particular case of interest in this chapter, we thus provide a basic roadmap through some of the most successful techniques.

- Wavelet footprints [9]: for piecewise polynomial signals, group together significant wavelets of pre-defined singularities into a footprint dictionary. The algorithm locates singularities and then selects the best footprints in the dictionary. In 1-D, it reaches the near optimal bound. In 2-D, situation gets complicated by the problem of chaining these footprints together along contours.

- Wedgeprints [10]: weds wavelets and wedgelets [11] by grouping wavelet coefficients into wedgeprints in the wavelet domain. One advantage is that all is computed based on the wavelet coefficients : they are sorted in a tree like manner according to their behavior as smooth, wedgeprints or textures. Markov trees help ensuring that particular grouping of coefficients do make sense (i.e., they represent smooth edges). It reaches the near optimal bound in 2-D.

- Bandelets [12]: the image is processed so as to find its edges. The wavelet transform is then warped along the geometry in order to provide a sparse expansion. It reaches the near optimal bound for all smooth edges.

Highly non-linear approximations

Another interesting way of achieving sparsity for low bit rate image coding is to turn to very redundant systems. In particular, we will now focus on the use of highly non-linear approximations in redundant dictionaries of functions.

Highly non-linear approximation theory is mainly concerned with the following question : given a collection $\mathcal{D}$ of elements of norm one in a Banach space[1] $\mathcal{H}$, find an exact N-sparse representation of any signal s :

$$s = \sum_{k=0}^{N-1} c_k g_k \,. \tag{8.2}$$

The equality in (8.2) may not need to be reached, in which case a N-term approximant $\tilde{s}_N$ is found :

$$\tilde{s}_N = \sum_{k=0}^{N-1} c_k g_k, \quad \|s - \tilde{s}_N\| \leq \epsilon(N)\,, \tag{8.3}$$

for some approximation error ϵ. Such an approximant is sometimes called (ϵ, N)-sparse.

The collection $\mathcal{D}$ is often called a dictionary and its elements are called atoms. There is no particular requirements concerning the dictionary, except that it should span $\mathcal{H}$, and there is no prescription on how to compute the coefficients c_k in eq. (8.2). The main advantage of this class of techniques is the complete freedom in designing the dictionary, which can then be efficiently tailored to closely match signal structures.

Our ultimate goal would be to find the best, that is the sparsest, possible representation of the signal. In other words, we would like to solve the following problem :

$$\text{minimize } \|c\|_0 \text{ subject to } s = \sum_{k=0}^{K-1} c_k g_{\gamma_k},$$

[1] A Banach space is a complete vector space B with a norm $\|v\|$, for more information please refer to [13].

where $\|c\|_0$ is the number of nonzero entries in the sequence $\{c_k\}$. If the dictionary is well adapted to the signal, there are high hopes that this kind of representation exists, and would actually be sparser than a nonlinear wavelet-based approximation. The problem of finding a sparse expansion of a signal in a generic dictionary $\mathcal{D}$ leads to a daunting NP hard combinatorial optimization problem. This is however not true anymore for *particular* classes of dictionaries. Recently, constructive results were obtained by considering incoherent dictionaries [14, 15, 16], i.e. collections of vectors that are not too far from an orthogonal basis. These results impose very strict constraints on the dictionary, but yield a striking improvement : they allow to solve the original NP hard combinatorial problem by linear programming. As we will now see, this rigidity can be relaxed when we turn to the problem of finding sufficiently good N-term approximants, instead of exact solutions to eq. (8.2).

In order to overcome this limitation, Chen, Donoho and Saunders [17] proposed to solve the following slightly different problem :

$$\text{minimize } \|c\|_1 \text{ subject to } s = \sum_{k=0}^{K-1} c_k g_{\gamma_k}.$$

Minimizing the ℓ_1 norm helps finding a sparse approximation, because it prevents diffusing the energy of the signal over a lot of coefficients. While keeping the essential property of the original problem, this subtle modification leads to a tremendous change in the very nature of the optimization challenge. Indeed, this ℓ_1 problem, called *Basis Pursuit* or BP, is a much simpler problem, that can be efficiently solved by Linear Programming using, for example, interior point methods.

Constructive approximation results for redundant dictionaries however do not abound, contrary to the wavelet case. Nevertheless, recent efforts pave the way towards efficient and provably good nonlinear algorithms that could lead to potential breakthroughs in multi-dimensional data compression. For illustration purposes, let us briefly comment on the state of the art.

Recently, many authors focused on incoherent dictionaries, or, equivalently, dictionaries whose coherence μ is smaller than a sufficiently small constant C (i.e., $\mu < C$), whereas the coherence of a dictionary $\mathcal{D}$ is defined as :

$$\mu = \sup_{\substack{i,j \\ i \neq j}} |\langle g_i, g_j \rangle| \,. \tag{8.4}$$

Coherence is another possible measure of the redundancy of the dictionary and eq. (8.4) shows that $\mathcal{D}$ is not too far from an orthogonal basis when its coherence is sufficiently small (although it may be highly overcomplete).

Let us first concentrate on a dictionary $\mathcal{D}$ that is given by the union of two orthogonal bases in $\mathbb{R}^N$, i.e., $\mathcal{D} = \{\psi_i\} \cup \{\phi_j\}$, $1 \leqslant i, j \leqslant N$. Building on early results of Donoho and Huo [14], Elad and Bruckstein have shown a particularly striking and promising result [15]: if $\mathcal{D}$ is the concatenated dictionary described above with coherence μ and $s \in \mathbb{R}^N$ is any signal with a sufficiently sparse representation :

$$s = \sum_i c_i g_i \text{ with } \|c\|_0 < \frac{\sqrt{2} - 0.5}{\mu}\,, \tag{8.5}$$

then this representation is the unique sparsest expansion of s in $\mathcal{D}$ and can be exactly recovered by Basis Pursuit. In other words, we can replace the original NP-hard combinatorial optimization problem of finding the sparsest representation of s by the much simpler ℓ_1 problem. These results have been extended to arbitrary dictionaries by Gribonval and Nielsen [16], who showed that the bound in eq. ((8.5)) can be refined to :

$$\|c\|_0 < \frac{1}{2}\big(1 + \frac{1}{\mu}\big)\,.$$

So far the results obtained are not constructive. They essentially tell us that, if a sufficiently sparse solution exists in a sufficiently incoherent dictionary, it can be found by solving the ℓ_1 optimization problem. Practically, given a signal, one does not know whether such a solution can be found and the only possibility at hand would be to run Basis Pursuit and check *a posteriori* that the algorithm finds a sufficiently sparse solution. These results also impose very strict constraints on the dictionary, i.e., sufficient incoherence. But this has to be understood as a mathematical artifice to tackle a difficult problem : managing dependencies between atoms in order to prove exact recovery of a unique sparsest approximation. When instead ones wants to find sufficiently good N-term approximants, such a rigidity may be relaxed as shown in practice by the class of greedy algorithms described hereafter.

Greedy algorithms: Matching Pursuit

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* [18]. Assuming that all atoms in $\mathcal{D}$ have norm one, we initialize the algorithm by setting $R_0 = s$ and we first decompose the signal as

$$R_0 = \langle g_{\gamma_0}, R_0 \rangle g_{\gamma_0} + R_1\,.$$

Clearly g_{γ_0} is orthogonal to R_1 and we thus have

$$\|R_0\|^2 = |\langle g_{\gamma_0}, R_0\rangle|^2 + \|R_1\|^2 \, .$$

If we want to minimize the energy of the residual R_1 we must maximize the projection $|\langle g_{\gamma_0}, R_0\rangle|$. At the next step, we simply apply the same procedure to R_1, which yields

$$R_1 = \langle g_{\gamma_1}, R_1\rangle g_{\gamma_1} + R_2 \, ,$$

where g_{γ_1} maximizes $|\langle g_{\gamma_1}, R_1\rangle|$. Iterating this procedure, we thus obtain an approximant after M steps:

$$s = \sum_{m=0}^{M-1} \langle g_{\gamma_m}, R_m\rangle g_{\gamma_m} + R_M \, ,$$

where the norm of the residual (approximation error) satisifies

$$\|R_M\|^2 = \|s\|^2 - \sum_{m=0}^{M-1} |\langle g_{\gamma_m}, R_m\rangle|^2 \, .$$

Some variations around this algorithm are possible. An example is given by the weak greedy algorithm [19], which consists in modifying the atom selection rule by allowing to choose a slightly suboptimal candidate:

$$|\langle R_m, g_{\gamma_m}\rangle| \geqslant t_m \sup_{g \in \mathcal{D}} |\langle R_m, g\rangle| \, , \quad t_m \leqslant 1 \, .$$

It is sometimes convenient to rephrase Matching Pursuit in a more general way, as a two-step algorithm. The first step is a selection procedure that, given the residual R_m at iteration m, will select the appropriate element of $\mathcal{D}$:

$$g_{\gamma_m} = \mathcal{S}(R_m, \mathcal{D}) \, ,$$

where $\mathcal{S}$ is a particular selection operator. The second step simply updates the residual :

$$R_{m+1} = \mathcal{U}(R_m, g_{\gamma_m}).$$

One can easily show that Matching Pursuit converges [20] and even converges exponentially in the strong topology in finite dimension (see [18] for a proof). Unfortunately this is not true in general in infinite dimension, even though this property holds for particular dictionaries [21]. However, DeVore and Temlyakov [19] constructed a dictionary for which even a good signal, i.e., a sum of two dictionary elements, has a very bad rate of approximation: $\|s - s_M\| \geqslant C M^{-1/2}$. In this case a very sparse representation of the signal exists, but the algorithm dramatically fails to recover it! Notice though, that this again does in no way rule out the existence of *particular* classes of very good dictionaries.

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 [22, 23] solves this problem by recursively orthogonalizing the set of selected atoms using a Gram-Schmidt procedure. 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.

In the same time, greedy algorithms offer constructive procedures for computing highly non-linear N-term approximations. Although the mathematical analysis of their approximation properties is complicated by their nonlinear nature, interesting results are emerging (see for example [24, 25, 26, 27]). Let us briefly illustrate one of them :

Theorem 1 *Let $\mathcal{D}$ be a dictionary in a finite or infinite dimensional Hilbert space and let $\mu : \max_{k \neq l} |\langle g_k, g_l \rangle|$ be its* coherence*. For any finite index set I of size card$(I) = m < (1+1/\mu)/2$ and any $s = \sum_{k \in I} c_k g_k \in$ span$(g_k,\ k \in I)$, Matching Pursuit :*

1. *picks up only "correct" atoms at each step ($\forall n,\ k_n \in I$);*
2. *converges exponentially*

$$\|f_n - f\|^2 \leq ((1 - 1/m)(1+\mu))^n \, \|f\|^2.$$

The meaning of this theorem is the following. Take a dictionary for which interactions among atoms are small enough (low coherence) and a signal that is a superposition of atoms from a subset $\{g_k, k \in I\}$ of the dictionary. In this case Matching Pursuit will only select those correct atoms, and no other. The algorithm thus exactly identifies the elements of the signal. Moreover, since Matching Pursuit is looping in a finite dimensional subset, it will converge exponentially to f. The interested reader will find in [25, 27] similar results for the case when the signal is not an *exact* superposition of atoms, but when it can be well *approximated* by such a superposition. In this case again, Matching Pursuit can identify those correct atoms and produce N-term approximants that are close to the optimal approximation.

The choice of a particular algorithm generally consists in trading off complexity and optimality, or more generally efficiency. The image compression scheme presented in this chapter 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

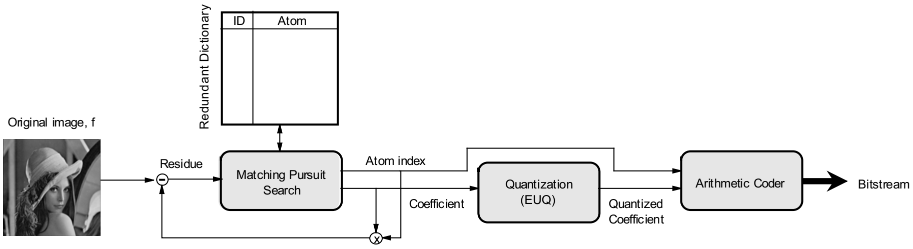


Figure 8.2: Block diagram of the Matching Pursuit image coder.

controlled complexity. Matching Pursuit, as already stressed before, iteratively chooses the best matching terms in a dictionary. Despite its possible numerical complexity in the signal representation, it is very easy to implement. Moreover, since there is almost no constraint on the dictionary itself, Matching Pursuit 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.

8.3 Matching Pursuit Image Coding

8.3.1 A Scalable Image Encoder

Overview

The benefits of redundant expansions in terms of approximation rate have been discussed in the first part of this chapter. The second part now describes an algorithm that builds on the previous results and integrates non-linear expansions over an anisotropically refined dictionary, in a scalable Matching Pursuit image encoder. The advantages offered by both the greedy expansion, and the structured dictionary are used to provide flexibility in image representation.

The encoder can be represented as in Figure 8.2. The input image is compared to a redundant library of functions, using a Matching Pursuit algorithm. Iteratively, the index of the function that best matches the (residual) signal is sent to an entropy coding stage. The corresponding coefficient is quantized, and eventually entropy coded. The output of the entropy coder block forms the compressed image bitstream. The decoder performs inverse entropy coding, inverse quantization, and finally reconstructs the compressed image by summing the dictionary functions, multiplied by their respective coefficients.

Clearly, the transform only represents one single stage in the compression chain. In order to take benefit of the improved approximation rate offered by redundant signal expansions, the quantization and entropy coding stage have also to be carefully designed. All the blocks of the compression algorithm have to be adapted to the specific characteristics of Matching Pursuit expansions. It is important to note that the real benefits of redundant transforms in image compression, can only be appreciated when all the blocks of the image encoder are fully optimized.

Alternative image representation methods based on Matching Pursuit, have been proposed in the literature. One of the first papers that proposed to use Matching Pursuit for representing images is [28]. This first work does however not propose a coder implementation, and the dictionary is different than the one proposed in this chapter. Matching Pursuit has been used for coding the motion estimation error in video sequences [29], in a block-based implementation. This coder, contrarily to the one proposed below, makes use of sub-blocks, which, in a sense, limits the efficiency of the expansion. In the same time, it has been designed to code the residual error of motion estimation, which presents very different characteristics than edge-dominated natural images. The coder presented in the remainder takes benefit of the properties of both redundant expansions, and anisotropic functions, to offer efficient and flexible compression of natural images.

Matching Pursuit Search

One of the well-known drawbacks of Matching Pursuit is the complexity of the search algorithm. The computations to find the best atom in the dictionary have to be repeated at each iteration. The complexity problem can be alleviated in replacing full search methods, by optimization techniques, such as implementations based on Tree Pursuit [30]. Although such methods greatly speed up the search, they often sacrifice in the quality of the approximation. They sometimes get trapped in local minima, and may choose sub-optimal atoms, which do not truly maximize the projection coefficient $|\langle g_\gamma | \mathcal{R}f \rangle|$. Other solutions can be found in efficient implementations of the Matching Pursuit algorithm, in taking benefit from the structure of the signal and the dictionary. The dictionary can for example be decomposed in incoherent blocks, and the search can thus be performed independently in each incoherent block, without penalty.

The actual implementation of the MP image encoder described here, still performs a full search over the complete dictionary, but computes all the projections in the Fourier domain [31]. This tremendously reduces the number of computations, in the particular case of our dictionary built on anisotropic refinement of rotated atoms. The number of multiplications in this case only depends on the number of scales and rotations in the dictionary, and does not depend any more on the number of atom translations.

The Matching Pursuit search in the Fourier domain allows to decrease the number of computations, possibly however at the expense of an increase in memory resources.

Generating functions of the dictionary

As presented in the previous section, a structured dictionary is built by applying geometric transformations to a generating mother function g. The dictionary is built by varying the parameters of a basis function, in order to generate 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, that is :

$$g(\vec{p}) = \frac{2}{\sqrt{3\pi}}(4\ x^2 - 2)\ \exp(-(x^2 + y^2))\ , \tag{8.6}$$

where $\vec{p} = [x, y]$ is the vector of the 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 [32].

The generating function described above is however not able to efficiently represent the low frequency characteristics of the image at low rates. There are two main options to capture these features: (i) to perform a low-pass filter of the image and send a quantized and downsampled image or (ii) to use an additional dictionary capable of representing the low frequency components. This second approach has also the advantage of introducing more *natural* artifacts at very low bit rate, since it tends to naturally distribute the available bits between the low and high frequencies of the image. A second subpart of the proposed dictionary is therefore formed by Gaussian functions, in order to keep the optimal joint space-frequency localization. The second generating function of our dictionary can be written as :

$$g(\vec{p}) = \frac{1}{\sqrt{\pi}} \exp(-(x^2 + y^2))\,, \tag{8.7}$$

where the Gaussian has been multiplied by a constant in order to have $||g(\vec{p})|| = 1$.

Anisotropy and orientation

Anisotropic refinement and orientation is eventually 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 :

$$\mathcal{D} = \{U(\gamma)g,\ \gamma \in \Gamma\}\ , \tag{8.8}$$

for a given set of indexes Γ. Basically this set must contain three types of operations :

- Translations $\vec{b}$, to move the atom all over the image.
- Rotations θ, to locally orient the atom along contours.
- Anisotropic scaling $\vec{a} = (a_1, a_2)$, to adapt to contour smoothness.

A possible action of $U(\gamma)$ on the generating atom g is thus given by :

$$U(\gamma)g = \mathcal{U}(\vec{b}, \theta)D(a_1, a_2)g \tag{8.9}$$

where $\mathcal{U}$ is a representation of the Euclidean group,

$$\mathcal{U}(\vec{b}, \theta)g(\vec{p}) = g\big(r_{-\theta}(\vec{p} - \vec{b})\big)\ , \tag{8.10}$$

r_θ is a rotation matrix, and D acts as an anisotropic dilation operator :

$$D(a_1, a_2)g(\vec{p}) = \frac{1}{\sqrt{a_1 a_2}} g\big(\frac{x}{a_1}, \frac{y}{a_2}\big)\ . \tag{8.11}$$

It is easy to prove that such a dictionary is overcomplete using the fact that, under the restrictive condition $a_1 = a_2$, one gets 2-D continuous wavelets as defined in [33]. It is also worth stressing that, avoiding rotations, the parameter space is a group studied by Bernier and Taylor [34]. The advantage of such a parametrization is that the full dictionary is invariant under translation and rotation. Most importantly, it is also invariant under isotropic scaling, e.g. $a_1 = a_2$. These properties will be exploited for spatial transcoding in the next sections.

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 $\vec{b}$, anisotropic scaling $\vec{a}$ and rotation θ. Any atom in our dictionary can finally be expressed in the following form :

$$g_\gamma = \frac{2}{\sqrt{3\pi}}(4\ {g_1}^2 - 2)\ \exp(-({g_1}^2 + {g_2}^2))\ , \tag{8.12}$$

with

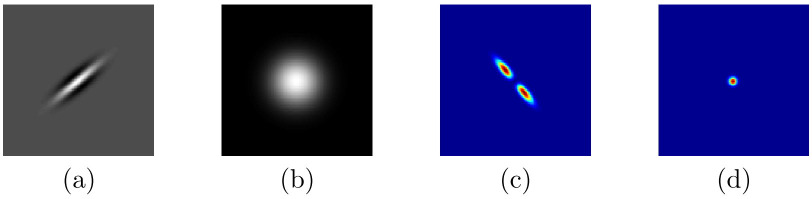
$$g_1 = \frac{\cos(\theta)\ (x - b_1) + \sin(\theta)\ (y - b_2)}{a_1}\ , \tag{8.13}$$

and

$$g_2 = \frac{\cos(\theta)\ (y - b_2) - \sin(\theta)\ (x - b_1)}{a_2}\ . \tag{8.14}$$

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. The rotation parameter varies by increments of $\frac{\pi}{18}$, to ensure the overcompleteness of the dictionary. 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 99 % 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, atoms are chosen to be 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. This limitation allows to limit the size of the dictionary.

For the Gaussian (low frequency) sub-dictionary, the translation parameters vary exactly in the same way as for the AR atoms, but the scaling is isotropic and varies from $\frac{min(W,H)}{32}$ to $\frac{min(W,H)}{4}$ on a logarithmic scale with a resolution of one third of octave (W and H are image width and height respectively). The minimum scale of these atoms has been chosen to have a controlled overlap with the AR functions, i.e., large enough to ensure a good coverage of the signal space, but small enough to avoid destructive interactions between the low-pass and the band-pass dictionary. This overlap has been designed so that less than 50% of the energy of the Gaussians lies in the frequency band taken by the AR fuctions. The biggest scale for these Gaussian atoms has been chosen so that at least 50% of the atom energy lies within the signal space when centered in the image. Lastly, due to isotropy, rotations are obviously useless for this kind of atoms. Sample atoms are shown in Fig. 8.3.



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

Figure 8.3: Sample anisotropic atom with a rotation of $\frac{5\times\pi}{18}$ radians and scales of 4 and 8 (a), Sample Gaussian function (b), and their respective transforms (c) and (d).

Coding Stage

Compact signal representations also necessitate an efficient entropy coding stage, to remove statistical redundancy left in the signal representation. This stage is crucial in overcomplete signal expansions, since the dictionary is inherently more redundant than in the case of common orthogonal transforms. Optimal coding in redundant expansions is however still an open research problem, that is made non-trivial by the large number of parameters in the case of image coding.

Efficient coding of Matching Pursuit parameters has been proposed in [29] for example, with a smart scanning of atom positions within image blocks. The coder presented in this section aims at producing fully scalable image streams. Such a requirement truly limits the options in the entropy coding stage, since the atom order is given by the magnitude of their coefficients, as discussed in the previous paragraph. The scalable encoder therefore implements an adaptive arithmetic coding, with independent contexts for position, scale, rotation and coefficient parameters. The core of the arithmetic coder is based on [35], with the probability update method from [36]. As the distribution of the atom parameters (e.g., positions or scales) is dependent on the image to be coded, the entropy coder first initializes the symbol probabilities to a uniform distribution. The encoded parameters are then sent in their natural order, which results in a progressive stream, that can eventually be cut at any point to generate rate scalable streams.

Recall finally that flexibility is the main motivation for choosing this kind of arithmetic coder. It can be imagined that more efficient coders could for example try to estimate the parameters distribution in order to optimally distribute the bits. Alternatively, grouping atoms according to their position parameters might also increase the compression ratio when combined with differential coding. Similar methods could be applied to rotation or scale indexes. However, the generated stream would not be progressive anymore, and scalability would only be attained in this case by stream manipulations, and more generally transcoding.

Coefficient Quantization

One of the crucial points in the Matching Pursuit encoder is the coefficient quantization stage. Since coefficients computed by the Matching Pursuit search take real values, quantization is a mandatory operation in order to limit the coding rate. Redundant signal expansions present the advantage that quantization error on one coefficient may be mitigated by later Matching Pursuit iterations, when the quantization is performed in the loop [37]. The encoder presented in this section however uses a different approach, that performs quantization *a posteriori*. In this case, the signal expansion does not depend on the quantization, and hence the coding rate. *A posteriori* quantization and coding allow for one single expansion to be encoded at different target rates. This is particularly interesting in scalable applications, which represent the main target for the image coder under consideration here. Since the distortion penalty incurred by a posteriori quantization is moreover generally negligible [38], this design choice is justified by an increased flexibility in image representation.

The proposed coder uses a quantization method specifically adapted to the Matching Pursuit expansion characteristics, the *a posteriori* rate optimized exponential quantization. It takes benefit from the fact that the Matching Pursuit coefficient energy is upper-bounded by an exponential curve, decaying with the coefficient order. The quantization algorithm strongly relies on this property, and the exponential upper-bound directly determines the quantization range of the coefficient magnitude, while the coefficient sign is reported on a separate bit. The number of quantization steps is then computed as the solution of a rate-distortion optimization problem [38].

Recall that the coefficient c_{γ_n} represents the scalar product $\langle g_{\gamma_n}, \mathcal{R}^n f \rangle$. It can be shown that its norm is upper-bounded by an exponential function [39], which can be written as

$$|c_{\gamma_n}| \leq (1 - \alpha^2 \ \beta^2)^{\frac{n}{2}} \, \|f\| \ . \tag{8.15}$$

where $\|f\|$ is the energy of the signal to code, β is a constant depending on the construction of the dictionary, and α is a sub-optimality factor depending on the Matching Pursuit implementation (for a full search algorithm as the one used in this paper, $\alpha = 1$). The coefficient upper-bound thus depends on both the energy of the input function and the construction of the dictionary. Since the coefficients can obviously not bring more energy than the residual function, the norm of the coefficient is strongly related to the residual energy decay curve.

Choosing the exponential upper-bound from Eq. (8.15) as the limit of the quantization range, it remains to be determined the number of bits to be spent on each coefficient. The rate-distortion optimization problem shows that the number of quantization levels have also to follow a decaying exponential law, given by :

$$n_j = \sqrt{\frac{\|f\|^2 \ (1-\beta^2)^j \ \log 2}{6 \ \lambda}} \ , \tag{8.16}$$

where n_j is the number of quantization levels for coefficient c_j, and λ is the Lagrangian multiplier which drives the size of the bit-stream [38].

In practice, the exponential upper-bound and the optimal bit distribution given by Eq. (8.16) are often difficult to compute, particularly in the practical case of large dictionaries. To overcome these limitations, the quantizer uses a suboptimal but very efficient algorithm based on the previous optimal results. The key idea lies in a dynamic computation of the redundancy factor β from the quantized data. Since this information is also available at the decoder, this one is able to perform the inverse quantization without any additional side information.

In summary, the coefficients quantization stage is implemented as follows. The coefficients are first re-ordered, and sorted in the decreasing order of their magnitude (this operation might be necessary since the MP algorithm does not guarantee a strict decay of the coefficient energy). Let then $Q[c_k]$, $k = 1, \dots j-1$ 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. (8.16). The adaptive quantization uses an estimate of the redundancy factor to compute the number of quantization levels as :

$$n_j = (1 - \tilde{\beta}_{j-1}^2)^{\frac{1}{2}} \ n_{j-1} \, . \tag{8.17}$$

The estimate of the redundancy factor $\tilde{\nu}$ is recursively updated, as :.

$$\tilde{\beta_j} = \sqrt{1 - \left(\frac{Q[c_j]}{\|f\|}\right)^{2/j}} . \tag{8.18}$$

Finally, the quantization range is given by the quantized coefficient $Q[c_j]$.

Rate Control

The quantization algorithm presented above is completely determined by the choice of n_0, the number of bits for the first coefficient, and a positive value of N, the number of atoms in the signal expansion. When the bitstream has to conform to a given bit budget, the quantization scheme parameters n_0 and N can be computed as follows. First, β is estimated with Eq. (8.18) by training the dictionary on a large set of signals (e.g., images), encoded with the adaptive quantization algorithm. The estimation quite rapidly tends to the asymptotic value of the redundancy factor. The estimation of β is then used to compute λ as a function of the given bit budget R_b which has to satisfy :

$$\begin{aligned} R_b &= \sum_{j=0}^{N-1} \log_2 n_j + \sum_{j=0}^{N-1} a_j \\ &= \sum_{j=0}^{N-1} \log_2 (1-\beta^2)^{\frac{j}{2}} + N \ \log_2 n_0 + N \ A \ , \end{aligned} \tag{8.19}$$

where a_j represents the number of bits necessary to code the parameters of atom g_{γ_j} (i.e., positions, scales and rotation indexes), and $A = E[a_j]$ represents the average index size. From Eq. (8.16), the value of λ determines the number of bits of the first coefficient n_0. Under the reasonable condition that the encoder does not code atoms whose coefficients are not quantized (i.e., $n_j < 2$) , the number of atoms to be coded, N is finally determined by the condition $(1-\beta^2)^{\frac{N-1}{2}} \ n_0 \leq 2$. The adaptive quantization algorithm is then completely determined, and generally yields bit rates very close to the bit budget.

8.3.2 Experimental Results

Benefits of anisotropy

Anisotropy and rotation represent the core of the design of our coder. To show the benefits of anisotropic refinement, our 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 :

$$U\big(a, \theta, \vec{b}\big) g(\vec{x}) = \frac{1}{a}\, g\big(a^{-1} r_{-\theta}(\vec{x} - \vec{b})\big)\,, \tag{8.20}$$

with

$$g(\vec{x}) = e^{i\vec{\omega_0}\cdot\vec{x}} e^{-\|\vec{x}\|^2/2}\,. \tag{8.21}$$

The next dictionary is an affine Weyl-Heisenberg dictionary [40] built by translation, dilation and modulation of the Gabor generating atom of Eq. (8.21) :

$$U\big(a, \vec{\omega}, \vec{b}\big) g(\vec{x}) = \frac{1}{a}\, e^{i\vec{\omega}\cdot(\vec{x}-\vec{b})} g\big(a^{-1}(\vec{x} - \vec{b})\big)\,, \tag{8.22}$$

where again, as we are dealing with real signals, only the real part is used. The other two dictionaries are simply built on orthogonal wavelet bases. Figure 8.4 shows the reconstructed quality 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 (the behavior of the coding rate is discussed below). Clearly, overcompleteness and anisotropic refinement allow to outperform the other dictionaries, in terms of approximation rate, which corresponds to the results presented in [4, 5]. 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.

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 8.5 illustrates the quality of the MP encoding of *Lena*, as a function of the coding rate, with both dictionaries. To perform the comparison, the isotropic and the anisotropic dictionaries are

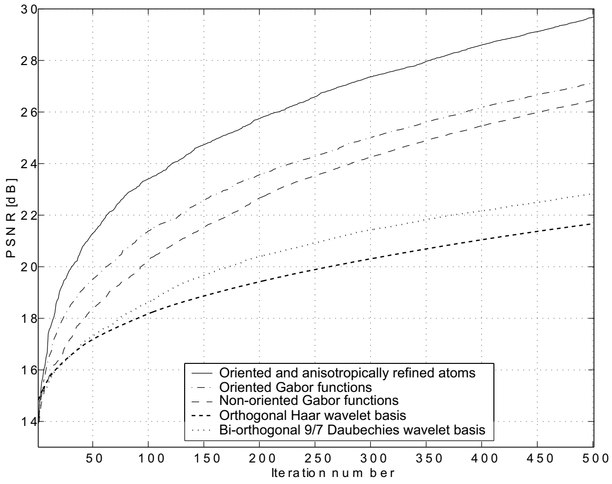


Figure 8.4: Comparison of the MP approximation rate for *Lena* (128 × 128 pixels), using five different dictionaries (anisotropic scaling, Gabor wavelets, Weyl-Heisenberg dictionary, an orthogonal Haar wavelet basis and a biorthogonal Daubechies 9/7 basis, with 5 levels of decomposition).

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.

Coding performance

The objective of this section is to emphasize the potential of redundant expansions low rate compression of natural images, even though the Matching Pursuit encoder is not fully optimized yet, as it has been discussed in Sec. 8.3.1.

Figure 8.6 presents a comparison between images compressed with Matching Pursuit, and respectively JPEG-2000[2]. It can be seen that the PSNR rating is in favor of JPEG-2000, which is not completely surprising since a lot of research efforts are being put in optimizing the encoding in JPEG-

[2] All results have been generated with the Java implementation available at http://jj2000.epfl.ch/, with default settings

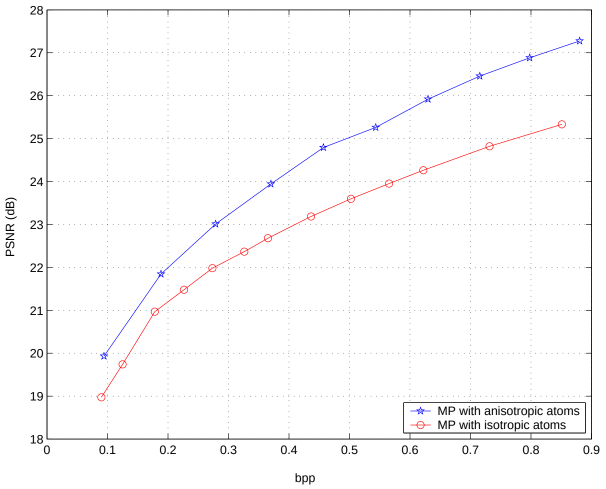
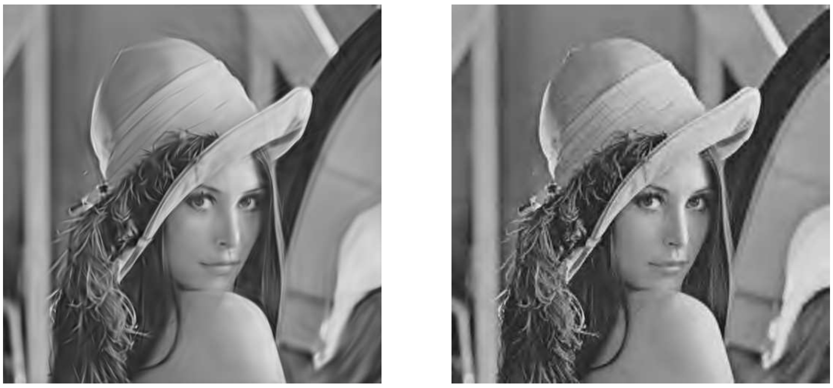


Figure 8.5: Comparison of the rate distortion characteristic of a decomposition using a dictionary built on anisotropic refinement, and a dictionary without anisotropic refinement. The basis functions are the same for isotropic and anisotropic functions, with the same angle discretization (allowing 18 different angles) and with spatial translation resolution of one pixel.

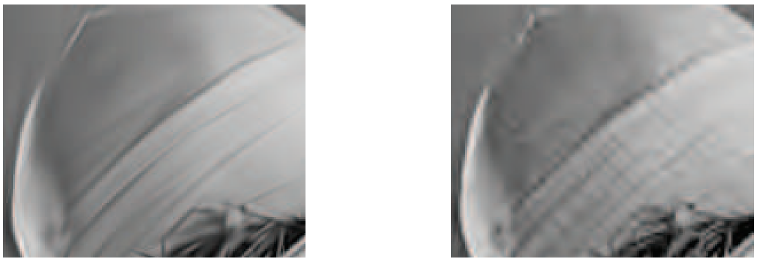
2000 like schemes. Interestingly, however, the image encoded with Matching Pursuit is visually more pleasant than the JPEG-2000 version. The coding artifacts are quite different, and the degradations due to Matching Pursuit are less annoying to the Human Visual System, than the ringing due to wavelet coding at low rate. The detailed view of the hat, as illustrated in Figure 8.7, confirms this impression. It can be seen that the JPEG-2000 encoder introduces quite a lot of ringing, while the MP encoder concentrates its effort on providing a good approximation of the geometrical components of the hat structure. JPEG-2000 has difficulties to approximate the 2-D oriented contours, which are generally the most predominant components of natural images. And this is clearly one of the most important advantages of the Matching Pursuit coder built on anisotropic refinement, which is really efficient to code edge-like features.

To be complete, Figure 8.8 shows the rate-distortion performance of the Matching Pursuit encoder for common test images, at low to medium bit rates. It can be seen that Matching Pursuit provides better PSNR rating than JPEG-2000 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 effi-



(a) MP, 31.0610dB (b) JPEG-2000, 31.9285 dB

Figure 8.6: *Lena* (512 x 512) encoded at 0.16bpp.

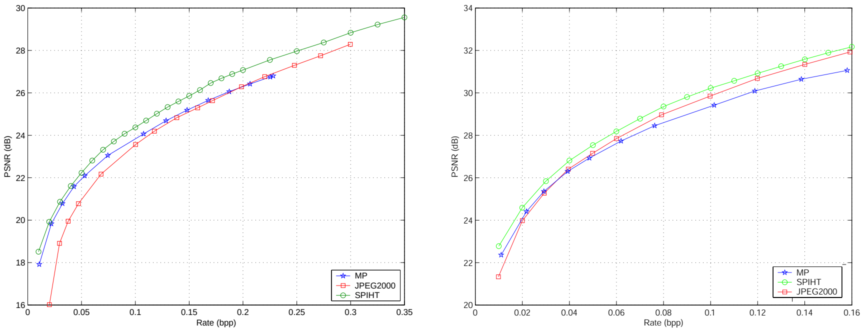


(a) MP (b) JPEG-2000

Figure 8.7: Detail view, *Lena* (512 x 512) encoded at 0.16bpp.

cient 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. It has to be noted also that the bitstream header penalizes JPEG-2000 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 the SPIHT encoder [41], 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 JPEG-2000 for high rates. However, the stream generated by the SPIHT encoder does in general not provide scalability, while MP and JPEG-2000 offer increased flexibility for stream adaptation.



Cameraman, 256 x 256 *Lena*, 512 x 512

Figure 8.8: Distortion-rate performance for JPEG-2000, SPIHT and the proposed MP coder, for common test images.

Finally, the proposed encoder performs reasonably well in terms of rate-distortion performance, especially at low rates. The distortion is in general visually less annoying in the Matching Pursuit coding algorithm. The artifacts introduced by Matching Pursuit (basically a simplification or refinable sketch of the image) are indeed less annoying for the human observer than the ringing introduced by the wavelets in JPEG-2000. When the rate increases, the saturation of the quality can be explained by the limitations of redundant transforms for high rate approximations. Hybrid coding schemes could provide helpful solutions for high rate coding.

8.3.3 Extension to color images

The Matching Pursuit encoder presented here-above can be extended to code color images, using similar principle. Instead of performing independent iterations in each color channel, a vector search algorithm can be implemented in a color image encoder. This is equivalent to using a dictionary of P vector atoms of the form $\{\vec{g_\gamma} = [g_\gamma, g_\gamma, g_\gamma]\}_{\gamma\in\Gamma}$. 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, or equivalently the vector component g_γ, with the highest energy. Hence the component of the dictionary chosen at each Matching Pursuit iteration satisfies:

$$\max_{\gamma_n} \sqrt{\langle R^n f^1, g_{\gamma_n}\rangle^2 + \langle R^n f^2, g_{\gamma_n}\rangle^2 + \langle R^n f^3, g_{\gamma_n}\rangle^2}, \tag{8.23}$$

where $R^n f^i$, $i = 1, 2, 3$ represents the signal residual in each of the color channels. Note that this is slightly different than the algorithm introduced in [42], where the sup norm of all projections is maximized :

$$\max_i \sup_{\mathcal{D}} |\langle R^n f^i, g_{\gamma_n} \rangle| \,.$$

All signal components f^i are then jointly approximated through an expansion of the from :

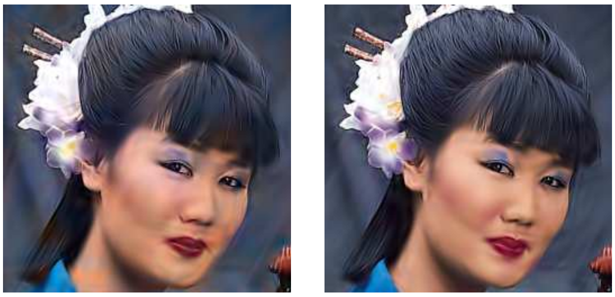
$$f^i = \sum_{n=0}^{+\infty} \langle R^n f^i, g_{\gamma_n} \rangle g_{\gamma_n}, \ \forall i = 1, 2, 3.$$

Note that channel energy is conserved, and that the following Parseval-like equality is verified :

$$\|f^i\|^2 = \sum_{n=0}^{+\infty} |\langle R^n f^i, g_{\gamma_n} \rangle|^2, \ \forall i = 1, 2, 3.$$

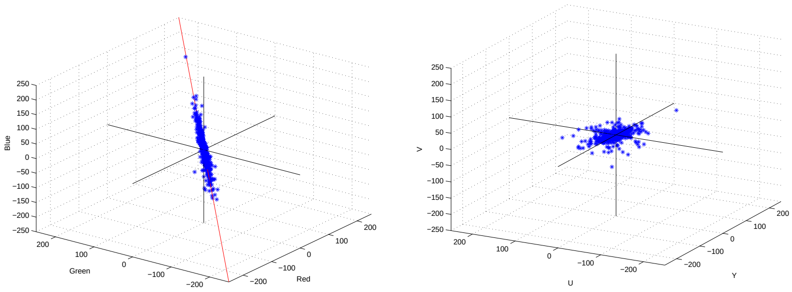
The search for the atom with the highest global energy necessitates the computation of the three scalar products $c_n^i = \langle R^n f^i, g_{\gamma_n} \rangle$, $i = 1, 2, 3$, for each atom g_{γ_n}, and for each iteration of the Matching Pursuit expansion. The number of scalar products can be reduced 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 obviously 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 (Fig. 8.9 is an example of a Matching Pursuit performed in the most energetic channel).

An important parameter of the color encoder is the choice of color space. Interestingly, experiments show that the MP coder tends to prefer highly correlated channels. This can be explained by the fact that atom indexes carry higher coding costs than coefficients. Using correlated channels basically means that the same structures are found, and thus the loss of using only one index for all channels is minimized. The choice of the RGB color space thus seems very natural. This can also be highlighted by the following experiments. 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 correspond to the red, green and blue components (see Fig. 8.10(a)). It can be seen that the MP coefficients are interestingly distributed along the diagonal of the color cube, or equivalently that the contribution of MP atoms is very



(a) YUV most energetic (b) RGB most energetic

Figure 8.9: Japanese woman coded with 1500 MP atoms, using the most energetic channel search strategy in YUV color space 8.9(a) and in RGB color space 8.9(b).

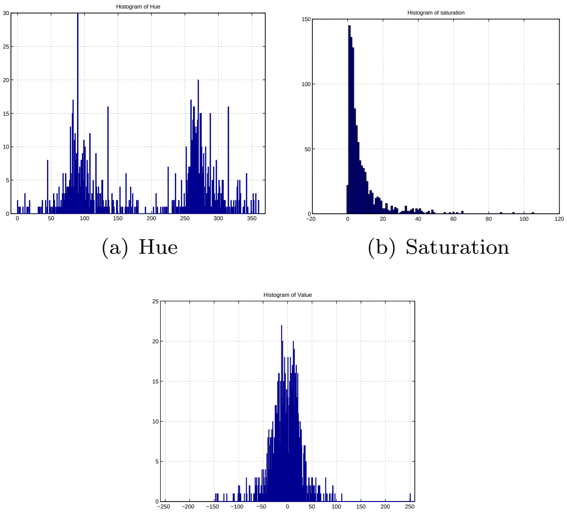


(a) RGB space (b) YUV space

Figure 8.10: Distribution of the MP coefficients, when MP is performed in the RGB or YUV color space.

similar in the three color channels. This very nice property is a real advantage in overcomplete expansions, where the coding cost is mainly due to the atom indexes. On the contrary, the distribution of MP coefficients, resulting from the image decomposition in the YUV color space, does not seem to present any obvious structure (see Fig. 8.10(b)). In addition, the YUV color space has been shown to give quite annoying color distortions for some particular images (see Fig. 8.9 for example).

Due to the structure of the coefficient distribution, centered around the diagonal of the RGB cube, efficient color quantization does not anymore consist in coding the raw values of the R, G and B components, but instead in coding the following parameters: the projection of the coefficients on the diagonal, the distance of the coefficients to the diagonal and the direction where it is located. This is equivalent to coding the Matching Pursuit 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



(a) Hue

(b) Saturation

(c) Value

Figure 8.11: Histograms of the MP coefficients when represented in HSV coordinates.

of the coefficient to the diagonal and H (Hue) is the direction perpendicular to the diagonal, where the RGB coefficient is located. The HSV values of the MP coefficients present the following characteristics distributions. The Value distribution is Laplacian, centered in zero (see Fig. 8.11(c)), Saturation presents an exponential distribution (see Fig. 8.11(b)), and a Laplacian-like distribution with two peaks can be observed for Hue values 8.11(a). Finally, once the HSV coefficients have been calculated from the available RGB coefficients, the quantization of the parameters is performed as follows:

- the Value is exponentially quantized with the quantizer explained before ([38]). The number 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.

- Hue and Saturation are uniformly quantized.

Finally coefficients and indexes are entropy coded, along the same technique used herebefore for grayscale images. Compression performances of this algorithm are illustrated on Figure 8.12, where a comparison with JPEG-2000 is also provided. It can be seen that MP advantageously compares to JPEG-2000, and even performs better at low bit rates. This can

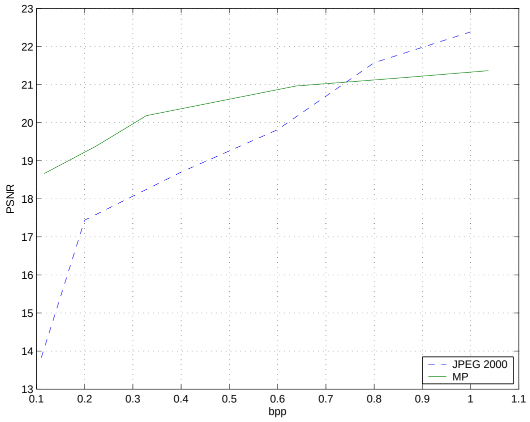


Figure 8.12: PSNR comparison between JPEG-2000 and MP. The PSNR has been computed in the CIELAB color space.

be explained by the property of MP to immediately capture most of the signal features in a very few iterations and across channels. Note that the PSNR values have been computed in the Lab color space, in order to match the Human Visual System perception.

8.3.4 High adaptivity

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 bitstreams, which ensure that the most important part of the information is available, independently of the number of bits used by the decoder [43, 44]. In order to enable easy rate adaptation, 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 bitstream can be decoded, even when truncated. On the other hand, efficient adaptive coding schemes, like JPEG-2000 or the coder proposed in [45] are generally based on subband 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 bitstreams 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 bitstream 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.

Spatial adaptivity

Due to the structured nature of our dictionary, the Matching Pursuit stream provides inherent spatial adaptivity. The group law of the similitude group of $\mathbb{R}^2$ indeed applies [33] and allows for invariance with respect to *isotropic* scaling of α, rotation of Θ and translation of $\vec{\beta}$. Let us remind the reader that the dictionary is built by acting on a mother function with a set of operators realizing various geometric transformations (see equations (8.8)-(8.9)). When considering only isotropic dilations, i.e. $a_1 = a_2$ in (8.9), this set forms a group : the similitude group of the 2-D plane. Therefore, when the compressed image $\hat{f}$ 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 :

$$\mathcal{U}(\eta)\hat{f} = \sum_{n=0}^{N-1} \langle g_{\gamma_n} | \mathcal{R}^n f \rangle \, \mathcal{U}(\eta) g_{\gamma_n} = \sum_{n=0}^{N-1} \langle g_{\gamma_n} | \mathcal{R}^n f \rangle \, \mathcal{U}(\eta \circ \gamma_n) g \, . \tag{8.24}$$

In the above expression $\gamma_n = (\vec{a}_n, \theta_n, \vec{b}_n)$ represents the parameter strings of the atom encoded at iteration n, with scaling $\vec{a}_n$, rotation θ_n and translation $\vec{b}_n$, 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 bitstream simply by modifying the parameter strings of the unit-norm atoms, according to the group law of similitude, where

$$\left(\vec{a},\theta,\vec{b}\right)\circ\left(\alpha,\Theta,\vec{\beta}\right)=\left(\alpha\cdot\vec{a},\theta+\Theta,\vec{b}+\alpha\cdot r_{\Theta}\vec{\beta}\right). \tag{8.25}$$

In other words, if $\eta_\alpha=(\alpha,0,0)$ denotes the isotropic scaling by a factor α, the bitstream 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 (from Eq. (8.25) and (8.9)). 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 :

$$\mathcal{U}(\eta_\alpha)\hat{f}=\alpha\sum_{n=0}^{N-1}c_{\gamma_n}g_{\eta_\alpha\circ\gamma_n}\,. \tag{8.26}$$

The modified atoms $g_{\eta_\alpha\circ\gamma_n}$ are simply given by Eq. (8.12) to (8.14), where $\vec{b}$ and $\vec{a}$ are respectively replaced by $\alpha\ \vec{b}$ and $\alpha\ \vec{a}$. 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 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 adaption procedure is illustrated in Fig. 8.13, 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 8.1 shows rate-distortion performance for spatial resizing of the 256×256 *Lena* image compressed at 0.3 bpp with the proposed

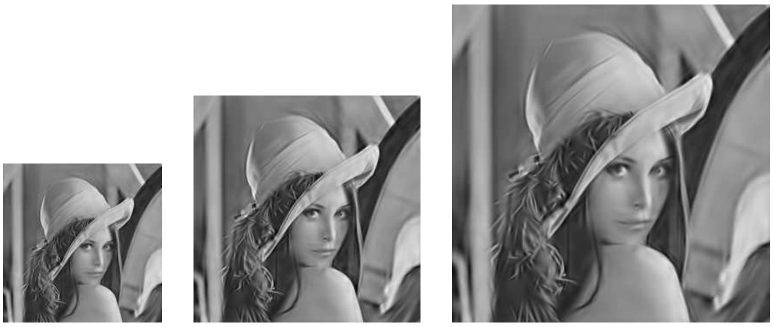


Figure 8.13: *Lena* image of size 256 × 256 encoded with MP at 0.3bpp (center), and decoded with scaling factors of $\sqrt{\frac{1}{2}}$ (left) and $\sqrt{2}$ (right).

Matching Pursuit coder, and JPEG-2000. It presents the PSNR values of the resized image, as well as the rate after transcoding. It also shows the PSNR values for encoding directly at the target spatial resolutions, for equivalent rates. The PSNR values have been computed with reference to the original 512 × 512 pixel *Lena* image, successively downsampled to 256 × 256 and 128 × 128 pixel resolutions. This is only one possibility for computing the low resolution reference images and other more complex techniques, involving for example filtering and interpolation, could be adopted. The choice of such a low resolution reference image was done in order not to favour one algorithm or the other. If a Daubechies 9/7 filter had bee chosen, JPEG would have given better results. On the contrary, if a Gaussian filter had been chosen, MP would have given better results. Note that the transcoding operations for JPEG-2000 are kept very simple for the sake of fairness; the high frequency subbands are simply discarded to get the lowest resolution images.

Table 8.1 clearly shows that our scheme offers results competitive with respect to state-of-the-art coders like JPEG-2000 for octave-based downsizing. In addition it allows for non-dyadic spatial resizing, as well as easy up-scaling. The quality of the down-scaled images are quite similar, but the JPEG-2000 transcoded image rate is largely superior to the MP stream one. 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. In the JPEG-2000 scenario however, the adaptation of the bitstream has a quite big impact on the quality of the reconstruction, compared to an encoding at the target resolution. Note

Encoder		128x128	256x256	512x512
Matching Pursuit	PSNR	27.34	30.26	27.5
	Rate [bpp]	0.8	0.3	0.08
	PSNR w/o tr.	27.4	30.26	27.89
JPEG-2000	PSNR	27.18	29.99	-
	Rate [bpp]	1.03	0.3	-
	PSNR w/o tr.	33.75	29.99	-

Table 8.1: Comparison of spatial adaptivity of the MP encoder and JPEG-2000. PSNR values are compared to quality obtained without transcoding (w/o tr.).

however that the PSNR value is highly dependent on the choice of the reference images, which in this case are simply downsampled from the original version.

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 bitstream, 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 Sec. 8.3.1. The encoded stream is then restricted to lower bit budgets r_k, $k = 0, \ldots, K$ by simply dropping the bits $r_k + 1$ to R_b. This simple rate-adption, or filtering operation is equivalent to dropping the last iterations in the MP expansion, focusing on the highest energy atoms.

Figure 8.14 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 bitstream has also been encoded directly at the different target rates r_k, as described in Sec. 8.3.1. 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 Sec. 8.3.1. 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

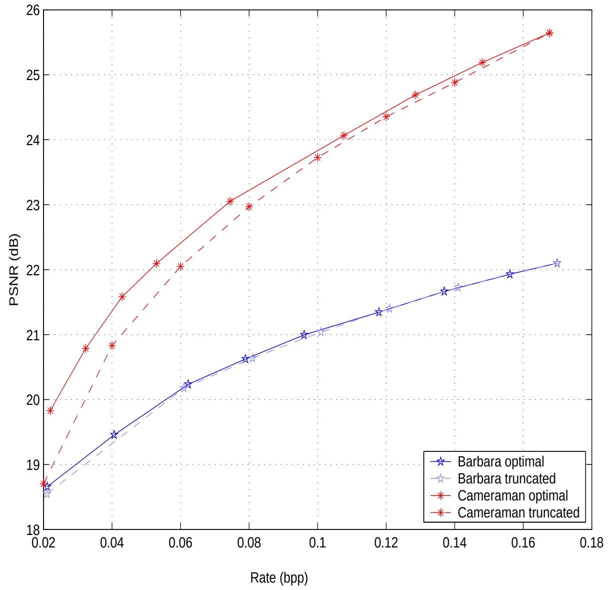


Figure 8.14: Rate-distortion characteristics for MP encoding of the 256 × 256 *Barbara* and *Cameraman* images at 0.17 bpp, and truncation/decoding at different (smaller) bit rates.

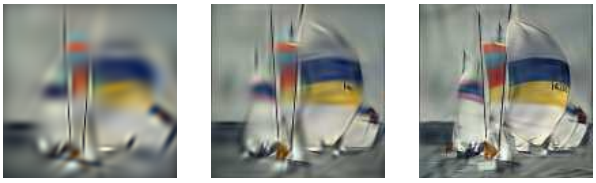
truncated rate-distortion curves are quite close, which shows that a simple rate adaption method, though quite basic, is very efficient.

Finally, rate scalability is also almost automatic for the color image stream. Fig. 8.15 shows the effects of truncating the MP expansion at different number of coefficients. It can be observed again that the MP algorithm will first describe the main objects in a sketchy way (keeping the colors) and then it will refine the details.

8.4 Discussions and conclusions

8.4.1 Discussions

The results presented in this chapter show that redundant expansions over dictionaries of non-separable functions may represent the core of new breakthroughs in image compression. Anisotropic refinement and orientation of dictionary functions allow for very good approximation rates, due to their ability to capture two-dimensional patterns, especially edges, in natural



(a) 50 coefficients (b) 150 coefficients (c) 500 coefficients

Figure 8.15: Matching Pursuit bitstream of *sail* image decoded after 50, 150 and 500 coefficients.

images. In addition, multidimensional representations may generate less annoying artifacts than wavelet or DCT transforms, that introduce ringing or blocking artifacts at low rates.

Matching Pursuit is however just one (sub-optimal) method that allows to solve the NP-hard problem of finding the best signal expansion in an overcomplete dictionary. It provides a computationally tractable solution with a very simple decoder implementation, and has the advantage to generate a scalable and progressive bitstream. Due to its greedy nature, Matching Pursuit however presents some limitations at high rate. An hybrid scheme could help in high bit rate coding, and provide a simple solution to the limitations of Matching Pursuit.

Finally, the image encoder presented in this chapter mainly aims at illustrating the potential of redundant expansions in image compression. It has been designed in order to provide scalable streams, and this requirement limits the encoding options that could be proposed, and thus possibly the compression efficiency. It is clear that the proposed Matching Pursuit encoder is not fully optimized, and that numerous research problems remains to be solved, before one can really judge the benefit of redundant transforms in image compression. The approximation rate has been proven to be better than the rate offered in the orthogonal transform case, but the statistics of coefficients in subband coding, for example, present a large advantage in terms of compression. It is thus too early to claim that Matching Pursuit image coding is the next breakthrough in terms of compression, but it already presents an very interesting alternative, with competitive quality performance and increased flexibility. The current coding scheme of the MP coefficients is not optimal, contrarily to the very efficient coding of wavelet coefficients in JPEG-2000. The advantage of the multidimensional decomposition in terms of approximation rate, is thus significantly reduced under a rate-distortion viewpoint. The design of better coding scheme, carefully adapted to the characteristics of the Matching Pursuit representation, however represents a challenging research problem.

8.4.2 Extensions and future work

One of the striking advantages of using a library of parameterized atoms is that the reconstructed image becomes parameterized itself: it is described by a list of geometrical features, together with coefficients indicating the "strength" of each term. This list can be manipulated, as explained in Section 8.3.4. But these features can be used to perform many other different tasks. For example, these features can be thought of as a *description* of the image and can thus be used for recognition or classification. The description could also be easily manipulated or altered to encrypt the image or to insert an invisible signature.

The ideas of using redundant expansions to code visual information could be further extended to video signals. In this case, the coder can follow two main design strategies, one based on motion estimation, the other based on temporal transform. In the first case, a Matching Pursuit encoder can be used to code the residue from the motion estimation stage. The characteristics of this residue are however quite different than the features present in natural images. The dictionary need to be adapted to this mainly high frequency motion noise. In the scenario where the coding is based on a temporal transform, Matching Pursuit could work with a dictionary of three-dimensional atoms, where the third dimension represent the temporal component. In this case, atoms live in a block of frames, and the encoder works very similarly to the image encoder.

8.5 Acknowledgments

The authors would like to thank Rosa M. Figueras i Ventura for her help in designing the image coder, and the production of the experimental results presented in this chapter.

References

[1] Taubman D. and Marcellin M., *JPEG2000: Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, Boston, November 2001.

[2] DeVore R. A., "Nonlinear approximation," *Acta Numerica*, vol. 7, pp. 51–150, 1998.

[3] Donoho D.L., Vetterli M., DeVore R. A. and Daubechies I., "Data compression and harmonic analysis," *IEEE Transactions on Information Theory*, vol. 44, pp. 391–432, 1998.

[4] Do M. N., Dragotti P.L., Shukla R. and Vetterli M., "On the compression of two dimensional piecewise smooth functions," in *IEEE International Conference on Image Processing (ICIP)*, 2001.

[5] Figueras i Ventura R. M., Granai L. and Vandergheynst P., "R-D analysis of adaptive edge representations," in *Proceedings of IEEE Intl. Workshop on Multimedia Signal Processing (MMSP02)*, St Thomas, US Virgin Islands, 2002.

[6] Candès E. J. and Donoho D. L., "Curvelets – a surprisingly effective nonadaptive representation for objects with edges," in *Curves and Surfaces*, L.L. *et al.* Schumaker, Ed., (Vanderbilt University Press, Nashville, TN, 1999.

[7] Starck J.-L., Candès E. J. and Donoho D. L., "The curvelet transform for image denoising," *IEEE Transactions on Image Processing*, vol. 11, pp. 670–684, 2002.

[8] Do M.N. and Vetterli M., "Contourlets: A directional multiresolution image representation," in *Proceedings of the IEEE International Conference on Image Processing*, Rochester, September 2002, vol. 1, pp. 357–360.

[9] Dragotti P.L. and Vetterli M., "Wavelet footprints: Theory, algorithms and applications," *IEEE Transactions on Signal Processing*, vol. 51, no. 5, pp. 1306–1323, May 2003.

[10] Wakin M., Romberg J., Choi H., and Baraniuk R., "Geometric methods for wavelet-based image compression," in *International Symposium on Optical Science and Technology*, San Diego, CA, August 2003.

[11] Donoho D.L., "Wedgelets: Nearly-minimax estimation of edges," *Annals of Stat*, vol. 27, pp. 859–897, 1999.

[12] Le Pennec E. and Mallat S.G., "Geometrical image compression with bandelets," in *Visual Communications and Image Processing 2003*, Touradj Ebrahimi and Thomas Sikora, Eds., Lugano, Switzerland, July 2003, SPIE, pp. 1273–1286, SPIE.

[13] Rudin W., *Real and Complex Analysis*, Mc Graw Hill, 1987.

[14] Donoho D. L. and Huo X., "Uncertainty principles and ideal atomic decompositions," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2845–2862, November 2001.

[15] Elad M. and Bruckstein A.M., "A generalized uncertainty principle and sparse representations in pairs of bases," *IEEE Transactions on Information Theory*, vol. 48, no. 9, pp. 2558–2567, September 2002.

[16] Gribonval R. and Nielsen M., "Sparse representations in unions of bases," Tech. Rep. 1499, IRISA, Rennes (France), 2003.

[17] Chen S., Donoho D. L. and Saunders M. A., "Atomic decomposition by basis pursuit," *SIAM J. Scientific Comp.*, vol. 20, pp. 33–61, 1999.

[18] Mallat S. G. and Zhang Z., "Matching Pursuits With Time-Frequency Dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, December 1993.

[19] DeVore R. and Temlyakov V. N., "Some remarks on greedy algorithms," *Adv. Comp. Math.*, vol. 5, pp. 173–187, 1996.

[20] Jones L., "On a conjecture of huber concerning the convergence of projection pursuit regression," *The Annals of Statistics*, vol. 15, pp. 880–882, 1987.

[21] Villemoes L. F., "Nonlinear approximation with walsh atoms," in *Surface fitting and multiresolution methods*, Schumaker Mhaut, Rabut, Ed., 1997.

[22] Pati Y. C., Rezaifar R. and Krishnaprasad P. S., "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decompositions," in *Proc. 27th Asilomar Conf. on Signals, Systems and Comput.*, 1993.

[23] Davis G. M., Mallat S. and Zhang, Z., "Adaptive time-frequency decompositions," *SPIE J. of Opt. Eng.*, pp. 2183–2191, 1994.

[24] Gilbert A. C., Muthukrishnan S. and Strauss M. J., "Approximation of functions over redundant dictionaries using coherence," in *Proc. 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2003.

[25] Gribonval R. and Vandergheynst P., "On the exponential convergence of matching pursuits in quasi-incoherent dictionaries," Tech. Rep. 1619, IRISA, Rennes (France), 2003.

[26] J A Tropp, "Just relax: Convex programming methods for subset selection and sparse approximation," Tech. Rep., ICES, University of Texas at Austin, Austin, USA, 2004.

[27] J A Tropp, "Greed is good: Algorithmic results for sparse approximation," Tech. Rep., ICES, University of Texas at Austin, Austin, USA, 2003.

[28] Bergeaud F. and Mallat S., "Matching pursuit of images," in *Proc. IEEE Int. Conf. Image Proc.*, Washington DC, October 1995, vol. I, pp. 53–56.

[29] Neff R. and Zakhor A., "Very Low Bit-Rate Video Coding Based on Matching Pursuits," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 158–171, February 1997.

[30] Jost P., Vandergheynst P. and Frossard P., "Tree-Based Pursuit," Tech. Rep. 2004.13, EPFL - Signal Processing Institute, July 2004.

[31] Figueras i Ventura R. M., Divorra Escoda O. and Vandergheynst P., "Matching pursuit full search algorithm," Tech. Rep., F-Group (LTS-2), ITS, EPFL, 2003.

[32] Marr D., *Vision*, Freeman, San Francisco, 1982.

[33] Antoine J.-P., Murenzi R. and Vandergheynst P., "Directional wavelets revisited: Cauchy wavelets and symmetry detection in patterns," *Applied and Computational Harmonic Analysis*, vol. 6, pp. 314–345, 1999.

[34] Bernier D. and Taylor K., "Wavelets from square-integrable representations," *SIAM Journal on Mathematical Analysis*, vol. 27, no. 2, pp. 594–608, 1996.

[35] Witten I.H., Neal R.M. and Cleary J.G., "Arithmetic Coding for Data Compression," *Communications of the ACM*, vol. 30, no. 6, pp. 520–540, June 1987.

[36] Duttweiler D.L.and Chamzas C., "Probability estimation in arithmetic and adaptive-huffman entropy coders," *Image Processing, IEEE Transactions on*, vol. 4 Issue: 3, pp. 237–246, Mar 1995.

[37] De Vleeschouwer C. and Zakhor, A., "In-loop atom modulus quantization for matching pursuit and its application to video coding," *IEEE Transactions on Image Processing*, vol. 12, no. 10, pp. 1226–1242, October 2003.

[38] Frossard P., Vandergheynst P., Figueras i Ventura R. M. and Kunt M., "A Posteriori Quantization of Progressive Matching Pursuit Streams," *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp. 525–535, February 2004.

[39] Mallat S., *A Wavelet Tour of Signal Processing*, Academic Press, 2 edition, 1999.

[40] Cvetkovic Z. and Vetterli M., "Tight Weyl-Heisenberg Frames in $l^2(Z)$," *IEEE Transactions on Signal Processing*, vol. 46, no. 5, pp. 1256–1259, May 1998.

[41] Said A. and Pearlman W.A., "Reversible image compression via multiresolution representation and predictive coding," in *Proceedings of the SPIE - Visual Communication and Image Processing*, 1993, vol. 2094, pp. 664–674, http://www.cipr.rpi.edu/research/SPIHT/.

[42] Lutoborski A. and Temlyakov V.N., "Vector greedy algorithms," *Journal of Complexity*, vol. 19, no. 4, pp. 458–473, 2003.

[43] Said A. and Pearlman W. A., "A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, June 1996.

[44] Taubman D. and Zakhor A., "Multirate 3-D Subband Coding of Video," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 572–588, September 1994.

[45] Woods J. W. and Lilienfield G., "A Resolution and Frame-Rate Scalable Subband/Wavelet Video Coder," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 9, pp. 1035–1044, September 2001.