Sparse Approximation Using M-Term Pursuits with Applications to Image and Video Compression

Adel Rahmoune, Pierre Vandergheynst, Pascal Frossard
{adel.rahmoune, pierre.vandergheynst, pascal.frossard}@epfl.ch
Signal Processing Institute ITS
École Polytechnique Fédérale de Lausanne EPFL
CH- 1015 Lausanne, Switzerland

Abstract

This paper introduces an algorithm for sparse approximation in redundant dictionaries, called the M-Term Pursuit (MTP), based on the matching pursuit approach (MP). This algorithm decomposes the signal into a linear combination of selected atoms, chosen to represent the signal components. The MTP algorithm provides adaptive representation for signals in any dictionary. The basic idea behind the MTP, is to partition the dictionary into L disjoint sub-dictionaries, each carrying some meaningful information. Then it iteratively finds a k-term approximation. During each iteration, M atoms, where $M \leq L$, are selected based on some thresholding parameter γ. A comparison against the matching pursuit algorithm is provided both from the theoretical and the practical point of views. The approximation performances of the MTP algorithm have shown to yield comparable results with those of the matching pursuit. Furthermore, it has a reduced computational complexity compared to MP.

Two applications have been developed using the MTP: (i) an image compression scheme and (ii) a video compression technique. Both schemes have demonstrated the abilities of the MTP algorithm to be used for compact representation and coding in either image or video compression fields. The MTP-based image coder yields comparable results with those of JPEG-2000 in terms of rate-distortion and scalability. Whereas for the MTP-based video coder, it is shown to provide interesting performances when compared to the state-of-the-art coding schemes, such as H.264, MPEG-4 or the scalable MP3D, in terms of rate-distortion.

Index Terms

Sparse approximation, M-Term Pursuits, Matching pursuit, Dictionary partition, progressive image compression, scalable video compression.

I. Introduction

Greedy approaches have gained impetus in sparse signal approximation, both in theory and practice, in many fields such as mathematics, statistics, signal processing, and pattern recognition, etc. One can easily enumerate dozens of applications based on such approaches. For instance, audio signal analysis using atomic decomposition, image and video representation and compression using rich wavelet-like and ridge-like dictionaries, feature extraction in images, classification, pattern recognition and machine learning, and so forth. Of course, without mentioning their early use in statistical analysis, such as the projection pursuit algorithm.

In this paper, we analyze a greedy approach, called the M-term pursuit (MTP) algorithm, which can be regarded as belonging to the framework of the matching pursuit algorithm [1], [2].

II. Approximation in redundant dictionaries

The problem of approximating functions using linear combinations of a small number waveforms or atoms, is known as the sparse approximation problem. To obtain a compact expansion of a function which contains complex structures, we must adapt our expansion to the various components of the function. Adaptive linear expansions can be used to extract information from signals. One can obtain an adaptive decomposition of a signal by expanding the signal into a sum of waveforms whose properties match those of the different signal structures, which can be time-frequency localization, scale-space or phase-space features. Such adaptive representations are important in

This work has been supported by the Swiss National Science Foundation under grant 2000-067859.02.

signal processing applications such as image/video compression and analysis. The atoms which have been used for our expansions are drawn from a large and redundant collection, called a dictionary.

It was proved that in a finite dimensional space, computing the optimal solution for the sparse approximation problem of any signal in any dictionary is an *NP-complete* problem, which motivates the use of sub-optimal greedy algorithms.

The matching pursuit (MP) algorithm is a greedy algorithm which computes function expansions by iteratively selecting dictionary atoms which best correlate to signal structures. An orthogonalized version of matching pursuit (OMP) was introduced later in order to improve the approximation performances on the expense of some added computational complexity.

Recently however, it was pointed out recently that the convex relaxation approach, such as the Basis Pursuit (BB) algorithm, can be used to solve the sparse representation problem under certain conditions [3], [4].

Both Matching Pursuit or its orthogonalized version were proved to be efficient algorithms to tackle the sparse approximation (or recovery) problem in any type of dictionaries. Their drawback lies in their computational complexity, even by including all the necessary optimizations, it is still computational demanding. Mainly because at each iteration, it selects *one* atom, which correlates the residual signal structure.

A greedy algorithm, called the M-Term Pursuit algorithm, is presented in this paper, where at each iteration a number of M atoms are selected to form a sub-space highly correlated with the signal components. Then, the atom selection process is followed by an orthogonal projection of the signal onto the span of these atoms to give the best approximation in this span.

This paper is organized as follows: Section III provides some basic concepts in matrix algebra and extensively used throughout the remaining of the paper. The MTP algorithm is entirely described and is presented in Section IV. The theoretical approximation performances of the MTP are compared against those of MP in Section V. Section VI extends the algorithm into Hilbert spaces, and illustrates the gain in computational complexity when compared to MP. An image representation and compression scheme based on MTP is fully described in Section VIII along with a comparison against the JPEG-2000 standard in terms of rate-distortion (R-D) behavior. Another application for progressive and scalable video representation and compression using the MTP algorithm is discussed in Section IX. Its R-D performances are very comparable with those of the state-of-the-art video coders, such as H.264 and MPEG-4. Finally, some conclusions and future directions are discussed in Section X to wrap the paper.

III. Mathematical Preliminaries

A. The Dictionary

We work in the finite dimensional complex space C^d, though the extension to the Hilbert space H is described in section VI. The Hermitian inner product for C^d is defined as $\langle ., . \rangle$, and the norm as $\|.\|_2$. The dictionary $\mathcal{D}$ is a collection of unit norm elementary signals, which are called *atoms*, and denoted as ϕ_ω, where ω belongs to an index set Ω, of cardinality $N \gg d$, as

$$\mathcal{D} = \{\phi_\omega : \omega \in \Omega\} \tag{1}$$

A usual notation is to represent the dictionary by a matrix Φ of size $d \times N$, where the columns of Φ correspond to the atoms and whose order is irrelevant in the matrix Φ. Strictly speaking, Φ is the synthesis matrix of the dictionary. Its analysis matrix is merely the conjugate transpose Φ^*.

The M-term approximation problem, which is also referred as *sparsity-constrained approximation*, is to provide the best approximation of a signal f using a linear combination of M atoms or fewer from the dictionary. Formally,

$$\min_{c \in C^\Omega} \|f - \Phi c\|_2 \qquad \text{Subject to} \qquad \|c\|_0 \leq M. \tag{2}$$

Provided that the input signal has no representation using fewer than M atoms,

A sub-dictionary $\mathcal{D}_i$ is defined as a set of atoms drawn from $\mathcal{D}$. The atoms are indexed by the set Λ, of cardinality N_i. These atoms define a synthesis matrix Φ_i, of size $d \times N_i$, and an analysis matrix Φ_i^*, which are both sub-matrices of Φ and Φ^* respectively.

B. The Coherence

An important parameter of the dictionary is the *coherence*, which is defined as the maximum inner product between any two different atoms:

$$\mu \stackrel{def}{=} \max_{\lambda \neq \omega} |\langle \phi_\omega, \phi_\lambda \rangle|. \tag{3}$$

When the coherence is large, the atoms are very correlated to each other, and if $\mu = 1$ it means that $\mathcal{D}$ contains at least two identical atoms. On the other hand, when the coherence is small we say that the dictionary is *incoherent* and in the extreme case μ vanishes in an orthonormal basis.

C. The Cumulative Coherence

A more accurate estimate of the dictionary elasticity is the *cumulative coherence* [4], [5]. It measures how much an atom can be correlated to a set of atoms. It is defined as:

$$\mu_1(m) \stackrel{def}{=} \max_{|\Lambda|=m} \max_{\omega \notin \Lambda} \sum_{\lambda \in \Lambda} |\langle \phi_\omega, \phi_\lambda \rangle|. \tag{4}$$

When μ_1 grows slowly, the dictionary is said to be incoherent.

Property 1: The cumulative coherence has the following properties:

- It generalizes the coherence: $\mu_1(1) = \mu$ and $\mu_1(m) \leq m\mu$.
- In an orthonormal dictionary, $\mu_1(m) = 0$ for all m.

D. The Dictionary Partition

A dictionary partition $\mathcal{P}(\mathcal{D})$ is a collection of L disjoint sub-dictionaries, $\mathcal{D}_i \subseteq \mathcal{D}$, where $L \leq d$ and in such a way that their union preserves the dictionary $\mathcal{D}$ integrity. Formally, it is defined as:

Definition 1: $\mathcal{P}(\mathcal{D}) = \{\mathcal{D}_i : i = 1 \dots L\}$ such that:

- $\mathcal{D} = \bigcup_{i=1}^{L} \mathcal{D}_i$,
- $\mathcal{D}_i \bigcap \mathcal{D}_j = \emptyset$ for $i \neq j$

In matrix notation, the concatenation of the sub-matrices Φ_i, in a row-wise manner, must produce the original matrix Φ, regardless of the order of the columns.

Of course, one should expect that a sub-dictionary $\mathcal{D}_i$ may carry some meaningful information. Two special cases are predominant for dictionary analysis. One case is when $\mathcal{D}_i$ is an orthonormal basis, in such a case the dictionary $\mathcal{D}$ is simply the union of L orthonormal bases. For example, a dictionary consisting of the union of a wavelet and a Fourier basis. This class of structured dictionaries has received lot of research efforts and let to very interesting and established results [3], [6], [7].

The other case, which is of significant importance also, is when $\mathcal{D}_i$ has a given coherence [1] between its own atoms, but *very small* coherence between atoms where each is in a distinct sub-dictionary. We call this type of dictionary partition an *incoherent partition*. Though, there is very limited work in this direction we think that it is as important as the previous case. In an incoherent-partitioned dictionary, if m selected atoms, where $m \leq L$, and each one pertains to a different sub-dictionary, then they are linearly independent and their cumulative coherence μ_1 is small.

E. Vector and Matrix Norms

It is helpful to remind some definitions about vector and matrix norms. Let x and A be a vector and a matrix respectively, p and q integers in the remaining of this section.

The l_p norm of a matrix is defined as:

$$\|x\|_p \stackrel{def}{=} (|x_1|^p + \dots + |x_n|^p)^{\frac{1}{p}}. \tag{5}$$

Two common special cases are

[1] We use the term coherence interchangeably to refer to either μ or μ_1 depending on the context

- when p tends to ∞, called the *max norm*, l_∞ norm of a vector is defined as:

$$\|x\|_\infty \stackrel{def}{=} \max\{|x_1|, \cdots, |x_n|\}. \tag{6}$$

- when $p = 0$, called the *quasi norm*, l_0 norm is defined as the number of non-zero elements in x.

The norm of a matrix is given by the $\|A\|_{p,q}$ as follows:

$$\|A\|_{p,q} \stackrel{def}{=} \max_{x\neq 0} \frac{\|Ax\|_q}{\|x\|_p} = \max_{\|x\|_p=1} \|Ax\|_q. \tag{7}$$

An interesting equality is

$$\|A\|_{p,q} = \|A^*\|_{q',p'} \tag{8}$$

where $1/p + 1/p' = 1$, if $p = 1$ then p' is ∞ and so true for (q, q').

Another quantity measure related to the norm is the restricted minimum $(2, \infty)$ of a matrix, which is defined as the following:

$$\beta \stackrel{def}{=} \min_{x\neq 0} \frac{\|Ax\|_\infty}{\|x\|_2}. \tag{9}$$

If A spans C^d then β is strictly positive and less than one. This quantity, introduced in [1] to prove the convergence of MP in finite dimensional spaces, gives a worst case performance for the atoms to capture the signal energy.

F. Singular Values of The Gram Matrix

Suppose that Φ_Λ, with $|\Lambda| = m$, is a given sub-dictionary of m linearly independent atoms. Then, the Gram matrix is simply $G = \Phi_\Lambda^* \Phi_\Lambda$, i.e. $G(\lambda, \omega) = \langle \phi_\lambda, \phi_\omega \rangle$.

Now, each singular value $\sigma^2(\Phi_\Lambda)$ satisfies the following bounds [4]

$$1 - \mu_1(m-1) \leq \sigma^2(\Phi_\Lambda) \leq 1 + \mu_1(m-1) \tag{10}$$

Proof: Using the Geršgorin Disc Theorem [8], it states that each eigenvalue of G, i.e. $\sigma^2(\Phi_\Lambda)$, lies in one of the m discs

$$\Delta_\lambda \stackrel{def}{=} \{z : |G(\lambda,\lambda) - z| \leq \sum_{\omega\neq\lambda} |G(\lambda,\omega)|\} \tag{11}$$

Since the atoms are normalized, it follows that $G(\lambda, \lambda) = 1$ and the sum is bounded by $\mu_1(m-1)$. ■

The inverse of the Gram matrix is given by $G^{-1} = \Phi_\Lambda^\dagger (\Phi_\Lambda^\dagger)^*$. Where $\Phi_\Lambda^\dagger$ is the *Moore-Penrose generalized inverse*. The singular values $\sigma^2((\Phi_\Lambda^\dagger)^*)$ of the generalized inverse are also bounded as follows:

$$\frac{1}{1 + \mu_1(m-1)} \leq \sigma^2((\Phi_\Lambda^\dagger)^*) \tag{12}$$

Remark 1: Another lower bound [5] for σ^2 of G^{-1} is $(1 - 2\mu_1(m-1))/(1 - \mu_1(m-1))$. However, it is valid only when $\mu_1(m-1) < 1/2$ in contrast to the one in Eq. 12, which holds for any μ_1.

IV. The M-Term Pursuit Algorithm in Finite Spaces

Let $\mathcal{P}(\mathcal{D})$ be an incoherent partition of the dictionary $\mathcal{D}$ having a synthesis matrix Φ, into L sub-dictionaries, each $\mathcal{D}_i$ is associated with its synthesis matrix Φ_i. Suppose that $L < d$.

Then the decomposition of any signal f is performed iteratively. At each iteration, the algorithm performs two steps: (i) the selection step, and (ii) the projection step. During the selection step, all the inner products between each atom in $\mathcal{D}$ and f are calculated. The quantities $\|\Phi^* f\|_\infty$ and $\|\Phi_i^* f\|_\infty$ designate the largest inner product magnitude in $\mathcal{D}$ and $\mathcal{D}_i$ respectively. Then, only *the atom* $\phi_i \in \mathcal{D}_i$, if any, which satisfies the thresholding condition:

$$\frac{\|\Phi_i^* f\|_\infty}{\|\Phi^* f\|_\infty} \geq \gamma \tag{13}$$

is selected for a fixed threshold γ. Now, the collection of the selected atoms Φ_Λ has a cardinality $m \leq L$.

The second step consists in an orthogonal projection $P^\perp$ of f onto the span of the atoms indexed in Λ. Then signal is updated by extracting the quantity $P^\perp f$ from f.

Before citing the theorem, let state the following lemma about the existence of Φ_Λ for a given γ.

Lemma 1: (Cardinality of Φ_Λ) Suppose that γ is used to select Φ_Λ in a single iteration s.t,

$$\Phi_\Lambda = \{\phi_\lambda : \phi_\lambda = \underset{\phi \in \Phi_i}{\arg}\,(\|\Phi_i^* f\|_\infty), \frac{\|\Phi_i^* f\|_\infty}{\|\Phi^* f\|_\infty} \geq \gamma\} \tag{14}$$

and let $m = |\Phi_\Lambda|$. Then m is upper-bounded as follows:

$$m < \frac{1}{\gamma^2}\left(\frac{\mu_1(m-1)}{\beta^2} + \frac{1}{\beta^2} - 1\right) + 1. \tag{15}$$

This upper bound on the cardinality should be $|\Phi_\Lambda| \leq L$, otherwise it is meaningless!

Proof: The best approximation of f on Φ_Λ is obtained through the orthogonal projection on the span of Φ_Λ. This projection ($P^\perp f$) is defined as:

$$P^\perp f = \Phi_\Lambda \Phi_\Lambda^\dagger f, \tag{16}$$

where

$$\Phi_\Lambda^\dagger = (\Phi_\Lambda^* \Phi_\Lambda)^{-1} \Phi_\Lambda^*. \tag{17}$$

Denote $x = \Phi_\Lambda^* f$ and the Gram matrix $G = \Phi_\Lambda^* \Phi_\Lambda$, one can easily verify that

$$\|P^\perp f\|_2^2 = x^* G^{-1} x, \tag{18}$$

Where the inverse of the Gram matrix G^{-1} is also defined as $\Phi_\Lambda^\dagger (\Phi_\Lambda^\dagger)^*$. This matrix is Hermitian, hence we can bound the approximation by [8],

$$\|P^\perp f\|_2^2 \geq \sigma_{min}^2((\Phi_\Lambda^\dagger)^*)\|x\|_2^2. \tag{19}$$

By using Eq 12, we get

$$\|P^\perp f\|_2^2 \geq \frac{1}{1+\mu_1(m-1)}\|x\|_2^2, \tag{20}$$

We know that the all admitted atoms in Φ_Λ, their corresponding inner product magnitudes satisfy $|x_i| \geq \gamma\|x\|_\infty$, we have $\|x\|_2^2 \geq (1 + (m-1)\gamma^2)\|x\|_\infty^2$,

Introducing this expression into Eq 20 gives

$$\|P^\perp f\|_2^2 \geq \frac{(1+(m-1)\gamma^2)}{1+\mu_1(m-1)}\|x\|_\infty^2. \tag{21}$$

To proceed, we combine bounds in Eq 9 and Eq 21 to get

$$\|P^\perp f\|_2^2 \geq \beta^2 \frac{(1+(m-1)\gamma^2)}{1+\mu_1(m-1)}\|f\|_2^2. \tag{22}$$

Since $P^\perp f$ is orthogonal projection, it must always satisfy $\|P^\perp f\|_2^2 \leq \|f\|_2^2$. By using Eq 22 and rearranging the terms, we get the upper bound 15 of m for a given γ.

■

Theorem 1: Let $\mathcal{P}(\mathcal{D})$ be a partition of $\mathcal{D}$ and f be any signal. Fix a threshold γ s.t,

$$\gamma \leq \frac{1}{\beta}\sqrt{\frac{\mu_1(m-1)}{(m-1)}} + \epsilon \tag{23}$$

Suppose that Φ_Λ exists with $m = |\Phi_\Lambda|$, and that the MTP returns an approximation a_m in a single iteration.

Then the approximation error is bounded by:

$$\|f - a_m\|_2^2 \leq \left(1 - \frac{\beta^2(1+(m-1)\gamma^2)}{1+\mu_1(m-1)}\right)\|f\|_2^2. \tag{24}$$

Proof: The m-term approximation a_m of f produces an error $f - a_m$ satisfying the triangular equality, i.e:

$$\|f - a_m\|_2^2 = \|f\|_2^2 - \|P^\perp f\|_2^2. \tag{25}$$

Because of the orthogonal projector. Now, by substituting the lower bound on $\|P^\perp f\|$ in Eq 22, we get the approximation error bound in Eq 24. ■

The previous established bound holds only for a single iteration. When the same algorithm is run for any J iterations to have an approximation with ($k \geq L$ atoms. In this case at each iteration, m_j atoms are selected according to a threshold γ_i, which can be either identical for all iterations or adaptive. Then the approximation error is bounded as follows,

Corollary 1: If in J iterations, the MTP algorithm returns an approximation with a_k with k atoms s.t, $k = \sum_{j=1}^{J} m_j$. Then the approximation error is bounded as:

$$\|f - a_k\|_2^2 \leq \|f\|_2^2 \prod_{j=1}^{J} \left(1 - \frac{\beta^2(1 + (m_j - 1)\gamma_i^2)}{1 + \mu_1(m_j - 1)}\right). \tag{26}$$

By induction and using the results of theorem 1, one can prove the bound in corollary 1.

The formal description of the M-term Pursuit algorithm is given below.

Algorithm 1 The MTP Algorithm

INPUT:

The signal f and the number of iterations J

OUTPUT:

The approximation A_J and residual $R^J f$

PROCEDURE

1. Initialize the residual signal $R^0 f = f$ and the approximation $A_0 = 0$, the set of atoms $\Phi_\Lambda = \emptyset$, and the iteration number $j = 1$.
2. For $i = 1$ to L
   Find $\phi_i = \arg_{\phi \in \Phi_i}(\|\Phi_i^* R^{j-1} f\|_\infty)$
   IF $\frac{|\langle R^{j-1} f, \phi_i \rangle|}{\|\Phi^* R^{j-1} f\|_\infty} \geq \gamma$
   Then $\Phi_\Lambda \leftarrow \Phi_\Lambda \cup \{\phi_i\}$
3. Determine the orthogonal projection $P^\perp$ onto the span of Φ_Λ
4. Compute the approximation and the residual:

$$\begin{aligned} A_j &\leftarrow A_{j-1} + P^\perp R^{j-1} f \\ R^j f &\leftarrow R^{j-1} f - P^\perp R^{j-1} f \end{aligned}$$

5. Increment j, empty Φ_Λ ($\Phi_\Lambda = \emptyset$), and go to step 2 if $j \leq J$.

Remark 2: The γ parameter can be regarded either as a thresholding operator or as a relaxation parameter, depending on the context. For example, the case of $\gamma = 1$ corresponds to the pure matching pursuit algorithm, since only one atom is selected. On the other extreme, when $\gamma = 0$, it means that all the L atoms are selected. Notice also that the afore-mentioned analysis does not include any indication of how to find a γ which guarantees the selection of m atoms. In practice however, two approaches can be used to tune γ:

- a top-down approach: starting from one ($|\Phi_\Lambda| = 1$), γ is slowly decreased until $|\Phi_\Lambda|$ reaches m,
- or a bottom-up approach: starting from zero ($|\Phi_\Lambda| = L$) γ is smoothly increased until $|\Phi_\Lambda|$ reaches m.

V. Approximation Performance

In this section, we discuss the implications of the error bound in theorem 1 and corollary 1 for sparse approximation. The effect of the coherence μ_1, the dictionary parameter β and the threshold γ are also discussed along with a comparison against the matching pursuit algorithm.

A. Comparison with MP

The error bound in equation 24 requires the knowledge of the cumulative coherence μ_1, whose computational complexity is much higher than that of μ in any dictionary. Thus, by using the fact that $\mu_1(m) \leq m\mu$, the approximation error can be further bounded as:

$$\|f - a_m\|_2^2 \leq \left(1 - \frac{\beta^2(1 + (m-1)\gamma^2)}{1 + (m-1)\mu}\right) \|f\|_2^2. \tag{27}$$

Let $h_1(m,\mu,\beta)$ be the decay factor in Eq. 27 defined by

$$h_1(m,\mu,\beta) = \left(1 - \frac{\beta^2(1+(m-1)\gamma^2)}{1+(m-1)\mu}\right). \tag{28}$$

and similarly, assume $h_2(m,\mu,\beta)$ is the decay factor of the iterative MTP with an identical γ_j equals to γ for all iterations, i.e.

$$h_2(m,\mu,\beta) = \prod_{j=1}^{J}\left(1 - \frac{\beta^2(1+(m_j-1)\gamma^2)}{1+(m_j-1)\mu}\right). \tag{29}$$

Recall that the matching pursuit algorithm [1] has an error bound of the m-term approximation given by

$$\|f - a_m\|_2^2 \leq \left(1-\beta^2\alpha^2\right)^m \|f\|_2^2. \tag{30}$$

and without the loss of generality, we assume that $\alpha = 1$ (i.e. MP selects iteratively the *best* atom) and define $h_{mp}(m,\beta)$ as:

$$h_{mp}(m,\beta) = \left(1-\beta^2\right)^m. \tag{31}$$

In all the following graphs, the threshold γ is chosen from the interval [0.4, 0.7]. Smaller values of γ accounts for some inefficiency in approximation performance. Whereas when γ is close to one, the MTP tends to behave like the MP, i.e. it takes a single atom per iteration. Another important assumption is the *existence of m atoms for the fixed threshold γ*, that must be satisfied by Eq. 15.

Some sample values for the parameter β have been chosen to characterize classes of dictionaries (e.g. 0.3, 0.4, 0.5, and 0.6). Different values are assigned to the coherence mu in dictionaries (0.05, 0.02, 0.01 and 10^{-6}), these values can indicate the global trend of the approximation error. With no surprise, one should expect that the smaller is the coherence, the better must be the approximation in general.

B. The Behavior of h_1 and h_{mp}

The error bounds for the MTP and MP have been plotted for different values of β and μ versus m according to the parameter setting described above. The graphs of h_1 and h_{mp} are displayed in Figures. 1, 2, 3(a), with the following observations:

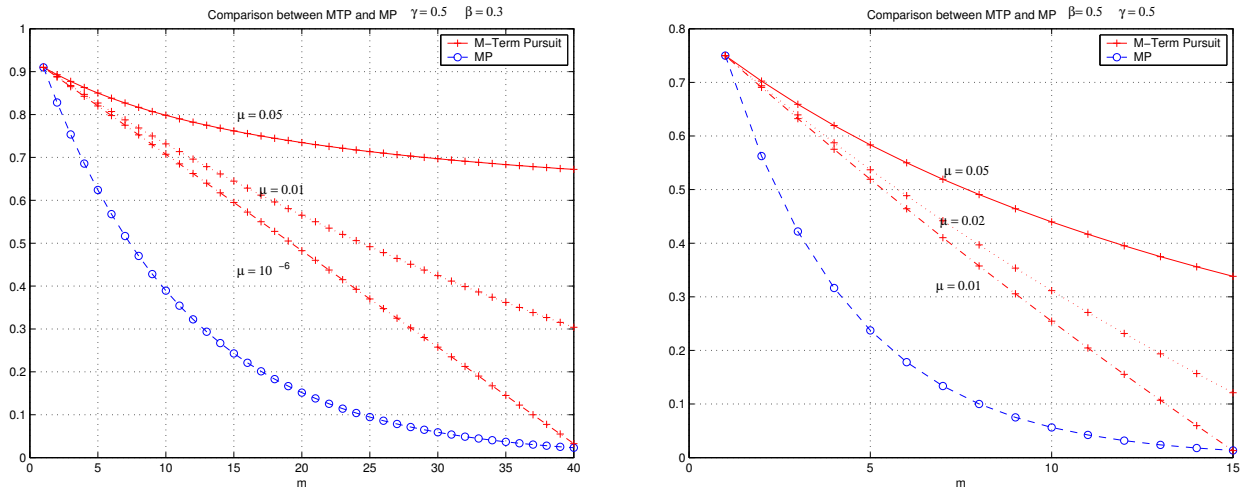
- In Fig.1, the error bound improves when the coherence μ decreases. And it takes fewer atoms for the MTP to achieve a comparable bound like that of MP when β is larger, (see the intersection point). For instance, when $\beta = 0.5$ and $\mu = 0.01$ the intersection occurs at m=15, whereas for the same μ and $\beta = 0.3$ it takes more than 40 atoms.
- We see also in Fig.2 that for a larger threshold γ, the error bound decreases, which is also intuitive, however the main issue lies in finding m atoms for that fixed threshold.
- Fig.3(a) shows the effect of β parameter when holding μ and γ fixed. We see clearly that the decay rate of the function h_1 is related to β. When β increases, the decay rate does so and vice versa.

C. The Behavior of h_2 and h_{mp}

The error bound function h_2 of the algorithm described in 1 with up to J iterations is compared to that of MP (h_{mp}). The graph in Figure. 3(b) is obtained for $\beta = 0.6$ and $\gamma = 0.5$ with a maximum number of atom selection ($|\Phi_\Lambda| = L$) equals to $L = 8$ per iteration.

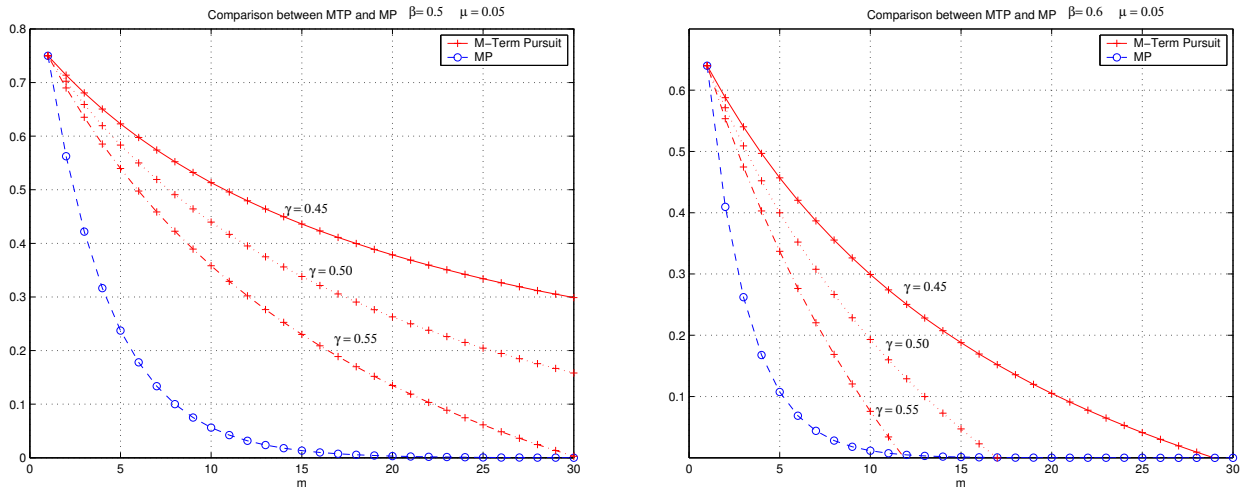
We see clearly jumps occurring between iterations. These discontinuities are due to the fact that h_1 has a fast decay rate with few atoms then tends to saturate. Such saturations are prevented by going through iterations. So in order to achieve a k-term approximation in any dictionary, a compromise should be found between the number of iterations J and the size of Φ_Λ in each iteration.

Remark 3: All the previous analysis on the error bound is performed on the theoretical worst-case scenario of $\mu_1(m)$, which would reach μm. This is very rough bound on μ_1. An accurate estimate of μ_1 should improve the error bound described in this section. In practice however, the performance of the algorithm is very interesting as we will see in section VII.



(a) $(\beta = 0.3, \gamma = 0.5)$ (b) $(\beta = 0.5, \gamma = 0.5)$

Fig. 1. Approximation error bounds comparison between h_1 and h_{mp} for different β's and γ's



(a) $(\beta = 0.5, \mu = 0.05)$ (b) $(\beta = 0.6, \mu = 0.05)$

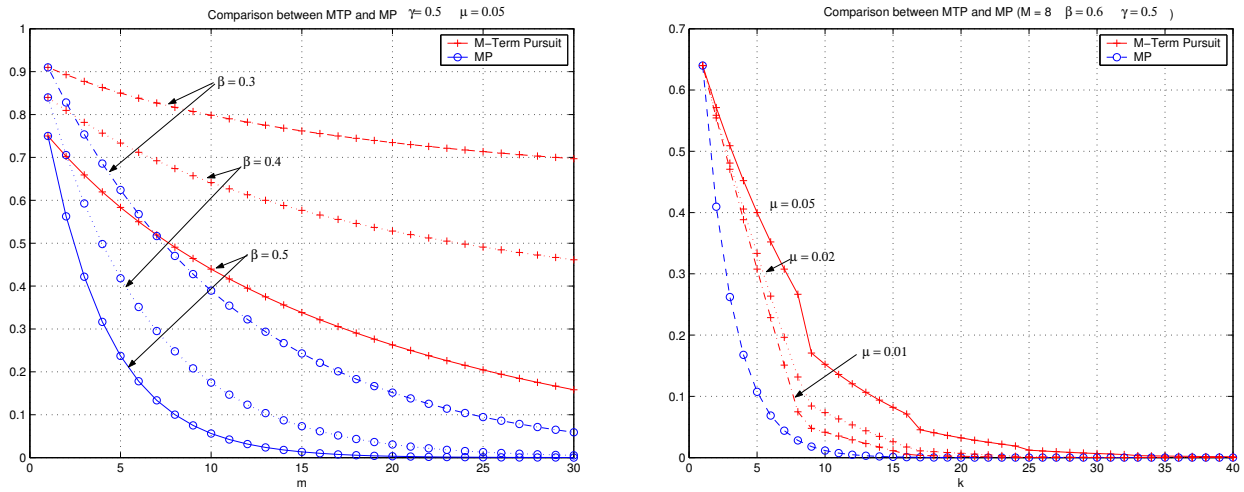
Fig. 2. Approximation error bounds comparison between h_1 and h_{mp} for different β's and μ's

VI. The M-Term Pursuit in Hilbert Spaces

The general issue is to expand a function f in the Hilbert space $H = L^2(R)$ over a dictionary $\mathcal{D} = \{g_\omega : \omega \in \Omega\}$, defined in H also. Interestingly, all the previously known concepts about dictionaries, dictionary partitioning, coherence and functions are still valid in this analysis. Though, the inner product between two functions is now defined in the Hilbert space H as

$$\langle f, g \rangle \stackrel{def}{=} \int_{-\infty}^{+\infty} f(t)\overline{g}(t)\, dt. \tag{32}$$

where $\overline{g}(t)$ is the complex conjugate of $g(t)$. An assumption is made on $\mathcal{D}$, is that the dictionary is dense and complete, i.e. it spans all the functions $f \in L^2(R)$. The main results of the M-Term Pursuit algorithm still applies to function expansion $f \in L^2(R)$ over $\mathcal{D}$.



(a) The effect of β on h_1 and h_{mp}

(b) Comparison between h_2 and h_{mp}

Fig. 3. Comparison of the error bounds between Iterative MTP and MP

A. The Approximation Algorithm

We define the dictionary partition $\mathcal{P}(\mathcal{D})$ in H, as previously, a collection of a finite L disjoint sub-dictionaries $\mathcal{D}_i = \{g_\omega : \omega \in \Omega_i\}$ where $\Omega_i \subseteq \Omega$. The selection of the atoms in the collection Φ_Λ is performed in a similar manner to the algorithm described in 2. The orthogonal projection P_V of f on the span of Φ_Λ is achieved by solving a linear system of equations. And after a single iteration of m-term approximation a_m, the residual error is given by

$$f - a_m = f - P_V f = f - \sum_{n=1}^{m} c_n g_{\omega_n} \tag{33}$$

Since $P_V f$ minimizes the error $\|f - \sum_{n=1}^{m} c_n g_{\omega_n}\|_2^2$ over all (c_n) for $n = 1 \cdots m$, the calculation of the coefficients (c_n) requires to solve the following linear system. For any $g_{\omega_k} \in \Phi_\Lambda$,

$$\langle f, g_{\omega_k} \rangle = \sum_{n=1}^{m} c_n \langle g_{\omega_n}, g_{\omega_k} \rangle. \tag{34}$$

Denote the vectors $C = (c_n)_{1 \le n \le m}$ and $X = (\langle f, g_{\omega_k} \rangle)_{1 \le k \le m}$, and the Gram matrix $G = (\langle g_{\omega_n}, g_{\omega_k} \rangle)_{1 \le k \le m, 1 \le n \le m}$. Once the linear system is solved in Eq. 34, or $GC = X$, we can verify that the residual energy of the approximation is

$$\|f - a_m\|_2^2 = \|f\|_2^2 - C^* X = \|f\|_2^2 - X^* G^{-1} X \tag{35}$$

If the atoms $g_{\omega_k} \in \Phi_\Lambda$ are linearly independent, then the quantity $X^* G^{-1} X$ is always positive and it is lower bounded by $\lambda_{min} \|X\|_2^2$. The minimum eigenvalue λ_{min} of the inverse Gram matrix G^{-1} is also lower bounded by $1/(1 + \mu_1(m-1))$ according to Eq. 12. Thus the energy extraction during each iteration is given by

$$X^* G^{-1} X \ge \frac{\|X\|_2^2}{1 + \mu_1(m-1)}, \tag{36}$$

The proof of the convergence of the MTP algorithm is identical to the proof of matching pursuit convergence [1], with the assumption that there exists α ($0 < \alpha \le 1$), s.t,

$$\|a_m\|_2 \ge \alpha \|\tilde{a}_m\|_2. \tag{37}$$

Where $\tilde{a}_m$ is the best m-term approximation for f in the dictionary $\mathcal{D}$.

Finally, the formal description of the algorithm is given in algorithm 2.

Algorithm 2 The MTP Algorithm in Hilbert Spaces

INPUT:

The signal f and the number of iterations J

OUTPUT:

The approximation a_J and residual $R^J f$

PROCEDURE

1. Initialize the residual signal $R^0 f = f$ and the approximation $a_0 = 0$, the set of atoms $\Phi_\Lambda = \emptyset$, and the iteration number $j = 1$.
2. For $i = 1$ to L

   Find $g_{\omega_i} = \arg\sup_{\omega \in \Omega_i} |\langle R^{j-1} f, g_\omega \rangle|$

   IF $|\langle R^{j-1} f, g_{\omega_i} \rangle| \geq \gamma \sup_{\omega \in \Omega} |\langle R^{j-1} f, g_\omega \rangle|$

   Then $\Phi_\Lambda \leftarrow \Phi_\Lambda \cup \{g_{\omega_i}\}$
3. Determine the orthogonal projection P_V onto the span of Φ_Λ
4. Compute the approximation and the residual:

$$\begin{aligned} a_j &\leftarrow a_{j-1} + P_V R^{j-1} f \\ R^j f &\leftarrow R^{j-1} f - P_V R^{j-1} f \end{aligned}$$

5. Increment j, empty Φ_Λ ($\Phi_\Lambda = \emptyset$), and go to step 2 if $j \leq J$.

B. The Computational Complexity

The computational complexity of the MTP algorithm is its main strength when compared to the matching pursuit algorithm, though it is thoroughly dependent on the dictionary. In one iteration, the MTP performs N inner products between each atom in D and the residual function $R^j f$. Assuming that each inner product requires p operations, the algorithm accumulates an order of $O(Np)$ operations. Once the m atoms are selected, it needs to solve the linear system $GC = X$. By using a Conjugate Gradient [9] solver, it requires a complexity of $O(qm)$, where q is a constant. Finally, the MTP culminates an order of $O(Np+mq)$ operations for m-term approximation in comparison to the $O(Nmp)$ complexity of the matching pursuit algorithm.

VII. APPLICATIONS

Two main applications are inherent to the M-Term Pursuit algorithm, (i) signal representation and compression and (ii) noise removal. The MTP algorithm is adaptive and greedy, at each iteration it searches for the m atoms that are highly correlated to the signal structures in sub-dictionaries. Once they are selected, an orthogonal projection is performed over these atoms to yield an m-term signal representation. Theoretical results in some worst case settings showed that the iterative step minimizes the error residual significantly which motivates to trade-off the number of iterations and the number of atoms selected during each iteration. Sparse signal representation is often associated with compression, so a step forward will be to apply the MTP algorithm to coding, hoping that the few selected m atoms will produce an efficient coding gain. An interesting class of compression is the progressive refinement coding, i.e. the quality of the reconstructed signal improves when more coded data arrives. It is very related to the MTP nature. At each iteration, the selected atoms and their coefficients are encapsulated and transmitted independently of the ulteriorly selected atoms. On the decoder side, upon reception of the bit-stream it continuously refines the reconstructed signal.

An application of the algorithm to image and video compression over some tuned visual dictionaries (aka spatio-temporal dictionaries) is described in the next sections along with comparison against other state-of-the-art approaches.

The other direction, which denoising or noise removal is entirely defined by the choice of the dictionary and the invariance measure of the algorithm map. However, it is not treated in the paper.

VIII. IMAGE REPRESENTATION AND COMPRESSION

Image coding and representation plays a crucial role in our daily life applications. For instance, multimedia compression and transmission over the Internet and mobile communication systems, searching and browsing in large

databases, detection in security-based systems, etc. All such tasks require a given decomposition or representation methodology, in our case the MTP algorithm, and a collection of image primitives or features which build up the dictionary $\mathcal{D}$. Once an image representation is obtained, all the previous tasks can be easily solved by applying a specific operator on this representation. In this section, we focus on scalable and progressive compression for images.

A. Image Dictionaries

Many approaches have been proposed to improve image representation (e.g. curvelets [10], bandelets [11], contourlets [12]) and all underline that an efficient image representation should have the following properties: (i) multi-resolution, (ii) localization: the basis functions should be localized in space and frequency (iii) directionality: the basis functions should be oriented at different directions (iv) anisotropic scaling: the basis functions should have a variety of elongated shapes with different aspect ratios.

For the spatial dictionary, we use the same construction as proposed in [13], that we sketch here for completeness. Two spatial mother atoms have been proposed, satisfying *the localization* property, a 2-D Gaussian and its 2^{nd} partial derivative,

$$g_1(x,y) = \frac{1}{\sqrt{\pi}} e^{-(x^2+y^2)}, \tag{38}$$

$$g_2(x,y) = \frac{2}{\sqrt{3\pi}} (4x^2 - 2) e^{-(x^2+y^2)}. \tag{39}$$

The 2-D Gaussian is used in order to extract the low frequency components and to generate a coarse approximation. Whereas the motivation behind using the 2^{nd} partial derivative of Gaussian, besides the localization property, is the need to have a function that efficiently captures image singularities like edges and contours.

The over-complete spatial dictionary is spanned by shifting, orienting, and scaling the spatial mother atoms using the following unitary operators :

- Shift:

$$\mathcal{U}_{(x_0,y_0)} g = g((x - x_0), (y - y_0)), \tag{40}$$

- Orientation:

$$\mathcal{U}_{\theta} g_2 = g_2(r_{-\theta}(x,y)), \tag{41}$$

- Scaling:

$$\mathcal{U}_a g_1 = \frac{1}{a} g_1(\frac{x}{a}, \frac{y}{a}), \tag{42}$$

$$\mathcal{U}_{(a_1,a_2)} g_2 = \frac{1}{\sqrt{a_1 a_2}} g_2(\frac{x}{a_1}, \frac{y}{a_2}), \tag{43}$$

For implementation issues, spatial position (x_0, y_0) sweeps the whole image and orientation may take 32 values $\theta = \frac{i\pi}{32}$, where $i = 0, \ldots, 31$. The scaling factor a_j, $j = 1, 2$, is logarithmically distributed as $a_j = 2^{\frac{i}{2}}$, with $i = 0, \ldots, 2\lfloor \log(\frac{size}{8}) \rfloor$.

B. Dictionary Partitioning

The dictionary $\mathcal{D}$ is partitioned into L sub-dictionaries $\mathcal{D}_i$. Each sub-dictionary $\mathcal{D}_i$ consists of all functions g's whose center, (x_0, y_0) defined in in equation 40, belongs to a region Ω_i. In our analysis, an image of size (X, Y) is divided into non-overlapping rectangular blocks of size (X_Ω, Y_Ω), where each block is associated to a region Ω_i and thus to a sub-dictionary $\mathcal{D}_i$. This class of partitioning, also called space-partitioning and denoted by $\mathcal{P}_{Omega}(\mathcal{D})$, insures that the coherence between atoms in different regions must be small, except for atoms lying near the border of the regions Ω_i's. The atoms lying near the border of a given region Ω_i generally have a support extending to other regions, though they are regarded to belong to Ω_i.

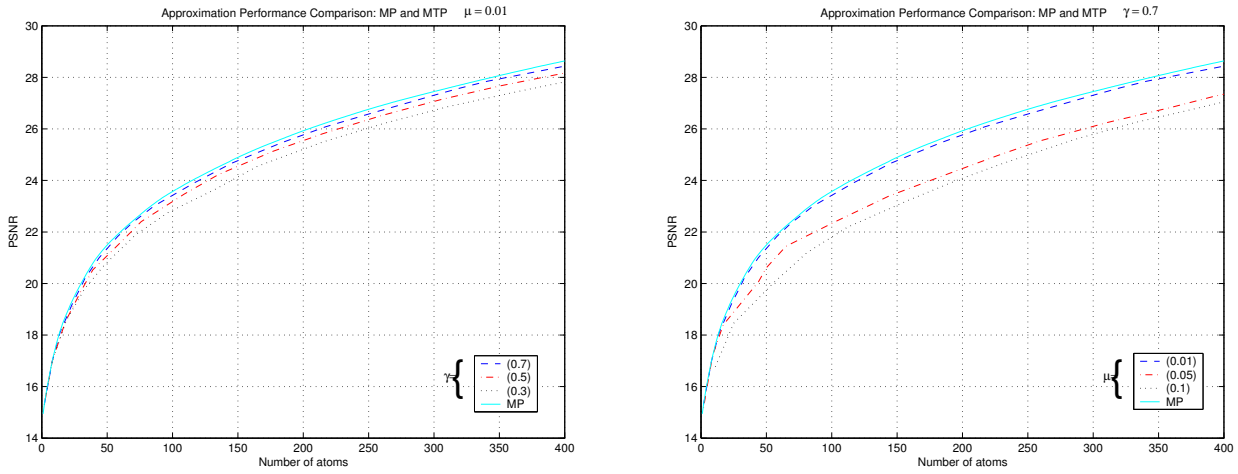
In order to guarantee that all selected atoms by the MTP algorithm possess small coherence, we defined a coherence threshold μ_s, such that no selected atom in Φ_Λ has a coherence $\mu_1(m)/m$ exceeding μ_s with the other atoms in Φ_Λ. In other words, every selected atom $\phi \in \Phi_\Lambda$ must satisfy the necessary condition of the coherence $\mu_1(m)/m \leq \mu_s$ in addition to the threshold condition, i.e. γ in equation 13.

Other dictionary partitioning based on either scales or orientations are still possible, and should be investigated as future directions.

C. Approximation Performance

The MTP algorithm has been applied on the dictionary $\mathcal{D}$ with the partition $\mathcal{P}_\Omega(\mathcal{D})$ as described in the previous subsection. Since the dictionary is invariant under translation, a fast implementation based on performing all the inner product between the image and the atoms in Fourier domain has a complexity of $O(n \log n)$, where n is the image size. The threshold γ is varied over $\{0.3, 0.5, 0.7\}$ and the coherence threshold μs is assigned also values from the set $\{0.01, 0.05, 0.1\}$. In all our experiments the dictionary $\mathcal{D}$ is divided into $L = 64$ sub-dictionaries $\mathcal{D}_i$.

Figure 4 shows the behavior of the error of MTP, measured in terms of the PSNR, versus the number of iterations compared against the matching pursuit algorithm for different γ's and μ_s's. It is clear that the gap between the two algorithms decreases with smaller μ_s and larger γ. With $\gamma = 0.7$ and $\mu_s = 0.01$ the MTP performs almost as well as the MP.



(a) ($\mu = 0.01$) (b) ($\gamma = 0.7$)

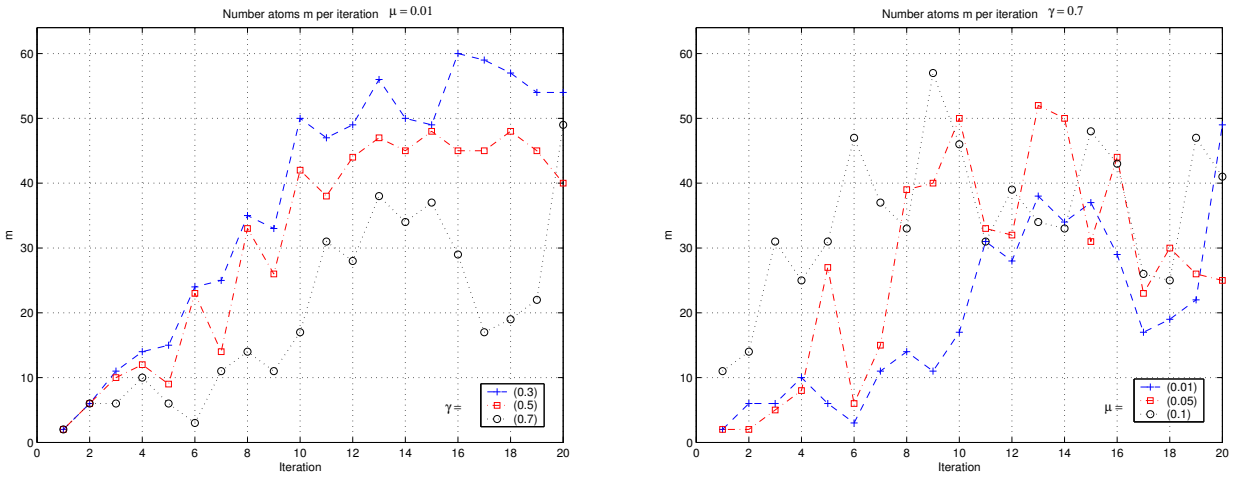
Fig. 4. Approximation performance comparison between MP and MTP (μ, γ) for Lena image (256x256)

An interesting relationship between the cardinality of Φ_Λ at each iteration and the values of the γ and μ_s for the Lena image is displayed in Figure 5. The number of selected atoms increases with the iterations, this is due to the fact that during the first iterations, most of the selected atoms correspond to the smooth regions present in the image, or the low-frequency part of the signal, which tend to have a support reaching over different regions Ω_i. Hence, these atoms with overlapping supports are characterized by somehow a *large* coherence. And by imposing a small coherence threshold μ_s on the atoms of Φ_Λ tends to reduce its cardinality. On the other hand, when iterations run on, the number of selected atoms per iteration stabilizes mainly because these atoms are chosen to represent edges, which are localized, and thus these atoms are almost uncorrelated. So their selection restriction comes principally from the fixed threshold γ.

A straightforward consequence of this observation is that the complexity gain of the MTP over MP, which is defined by the cardinality of Φ_Λ 14, becomes significant after few iterations.

D. Progressive Coding

The embedded quantization and coding is performed by using the subset approach introduced in [14], where the selected atoms (indexes and coefficients) are initially divided into S disjoint subsets s_i. Each subset contains l_i elements as shown in Figure 6. These subsets can be seen as energy sub-bands. Their number is dictated by scalability requirements (i.e., the number of target decoding rates), and represents a trade-off between stream flexibility, and coding efficiency, that respectively increases and decreases with S. In each subset, atoms are sorted according to their spatial positions, that are further run-length encoded. Other index parameters and quantized coefficients are encoded with an adaptive arithmetic encoder [15]. The resulting bit-stream is now piecewise progressive, and optimal truncation points can be set at subset limits. The rate control problem belongs to a general class of bit



(a) ($\mu = 0.01$) (b) ($\gamma = 0.7$)

Fig. 5. The effect of μ and γ on the number of atoms selected per iteration for Lena image (256x256)

allocation problems under multiple rate constraints, and is discussed in more details in the MP3D scheme [14]. Moreover, the embedded bit-stream allows for non-octave spatial scaling, due to the structured dictionary, without

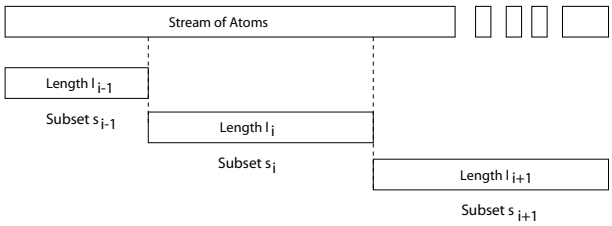


Fig. 6. The series of atoms is divided into energy subsets, as a compromise between coding efficiency and flexibility.

resorting to complex transcoding operations [14].

E. Rate-Distortion Results

In this subsection, we evaluate the rate-distortion performances of the MTP based image compression scheme and compare it against the state-of-the-art image compression standard JPEG-2000. The standard Lena image is used for comparison in our experiment. It can be seen on Figure 7 that the PSNR of MTP image codec is higher than that of JPEG-2000 by about 0.6 db over the range of low and medium bit rates, i.e. up to 0.4 bpp.

Figure 8 show the visual quality of the MTP-based codec when applied upon the Lena image. When decoded at a target bit rate is 0.41 bpp, it has a PSNR of 32.73. And one can see that almost all the edges and contours present in the image have been captured, which imparts some smoothness to the reconstructed image.

IX. Video Representation and Compression

Another important application prevalent in almost all multimedia-based services is video compression and delivery over the Internet and wireless networks, remote browsing in movie databases for video on demand services, etc.

Video representation is the first operation block used in any compression algorithm or video processing technique. Representing the video sequence as a linear combination of spatio-temporal waveforms, or atoms aligned along motion trajectories, requires (i) a decomposition algorithm, and (ii) a dictionary $\mathcal{D}$ of motion-adaptive spatio-temporal atoms.

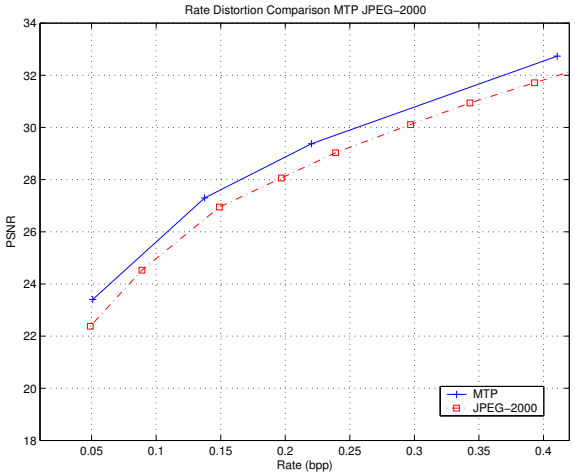


Fig. 7. Rate distortion comparison between MTP($\gamma = 0.7, \mu = 0.01$),and JPEG-2000 for Lena image (256x256)

(a) Original Lena

(b) MTP-based Codec

Fig. 8. The Lena image decoded using the MTP-based codec at 0.41bpp, the PSNR = 32.73

The MTP algorithm is used as a decomposition algorithm, and the dictionary $\mathcal{D}$ consists of spatio-temporal atoms mapped along motion trajectories, having spatial component as described in subsection VIII-A and a temporal component able to capture the temporal signal evolution.

A forward step for video representation is progressive compression, which has gained more focus recently, for ubiquitous applications. The aim of progressive coding is to generate a single embedded bit-stream able to be decoded and reconstructed efficiently over a given range of bit-rate and at various display formats.

The MTP is an iterative algorithm, where at each iteration an approximation to the residual signal is generated. So, the MTP progressively refines the approximation by going through more iterations and by adding more representative atoms. Now, the nature of the MTP and the use of the subsets-based embedded quantization and coding yields a highly flexible bit-stream well suited for scalable compression applications.

In this section, an overview of the dictionary is given subsection IX-A, the dictionary partitioning is described in subsection IX-B. The motion-adaptive MTP-based transform is presented in Subsection IX-C. The algorithm comprises two steps which are: (i) the motion trajectory prediction and (ii) MTP decomposition. A comparison against the MP based decomposition in terms of PSNR versus number of atoms is given in subsection IX-D for the sake of validation. A brief reminder of the embedded quantization and coding is presented in subsection IX-E. Experiments carried out on standard test sequences illustrate its coding efficiency in subsection IX-F. At low and medium bit rates, (i.e. less than 500 kbps), the obtained results are very comparable to the state-of-the-art

non-scalable coders H.264 and MPEG-4, and to our previous scalable MP3D scheme [14].

A. Spatio-temporal Dictionaries

The spatio-temporal dictionary, which consist of separable spatial and temporal components, has a structure able to efficiently represent the spatial image content according to subsection VIII-A, as well as temporal evolution along motion trajectories.

The temporal functions on the other hand should satisfy the following objectives. They should capture most of the signal energy in the low-pass temporal frequencies with few elements, as this will reduce the ghosting artifacts at low bit rates. They should satisfy multi-resolution and localization properties in order to encompass the wide range of temporal behavior observed in natural scenes. These properties are achieved by selecting a β-spline $\beta^n(t)$ function [16]. Trading-off between temporal and frequency decay, the order of $\beta^n(t)$ should be $n \geq 2$. While testing n, the 3^{rd} order β-spline $\beta^3(t)$ resulted in good performance for a GOP size of 16. The temporal part of the dictionary is thus generated by shifting and scaling the β-spline

$$\mathcal{T}_{t_0,s}\beta^3 = \beta^3(\frac{(t-t_0)}{s}). \tag{44}$$

The atom center t_0 sweeps the entire GOP size and the scale $s = 2^i$ varies according to $i = 0, \ldots, \lfloor \log(GOP) \rfloor$. It is noteworthy that in the temporal scale $s = 2^i$, i refers to the resolution or the number of frames that are processed in the signal. When $i = 0$, only 1 frame is processed, which can be interpreted as the existence of an abrupt motion, a scene change or an isolated feature. Whereas for $i = 1$, the support size is of 3 frames. It means that there is a smooth temporal evolution in the direction of motion localized in 3 frames. More generally, the larger the support size, the longer the motion trajectory.

To summarize, the redundant spatio-temporal dictionary is built from applying the coupled operators $\mathcal{U}$ and $\mathcal{T}$ on the 3-D mother atoms, along motion trajectories, in order to take advantage of the nature of the video signal.

B. Spatio-temporal Dictionary Partitioning

The spatio-temporal dictionary $\mathcal{D}$ is partitioned based on its spatial component into L sub-dictionaries $\mathcal{D}_i$, according to the method described in the previous section VIII-B (by using space-based partition) but with a slight modification. Let $I(x, y, t_0)$ be the t_0'th frame in the GOP, denoted as a *reference* frame. We define a dictionary partition $\mathcal{P}_{\Omega(t_0)}(\mathcal{D})$ as the set of L sub-dictionaries $\mathcal{D}_i$. Each $\mathcal{D}_i$ contains the spatio-temporal atoms $\mathcal{W}(g_\gamma(x_0, y_0, t_0))$ centered at the reference frame and such that (x_0, y_0) belongs to a region in the reference frame t_0 (or $(x_0, y_0) \in \Omega_i(t_0)$).

Similarly to the spatial dictionary partition in section VIII-B, the reference frame t_0 of size (X, Y) is divided into non-overlapping rectangular blocks of size (X_Ω, Y_Ω), where each block is associated to a region $\Omega_i(t_0)$. And to insure that all selected atoms by the MTP algorithm possess small coherence, we defined a coherence threshold μ_s, such that no selected atom in Φ_Λ has a coherence $\mu_1(m)/m$ exceeding μ_s with the other atoms in Φ_Λ.

C. The MTP Video Decomposition

The MTP algorithm decomposes the video signal in a finite number of spatio-temporal atoms along motion trajectories and it is described by Algorithm 3. The iterative algorithm first selects the frame with the highest energy, in the GOP $I^n(x, y, i)$, where $I^0(x, y, i)$ represents the original signal.

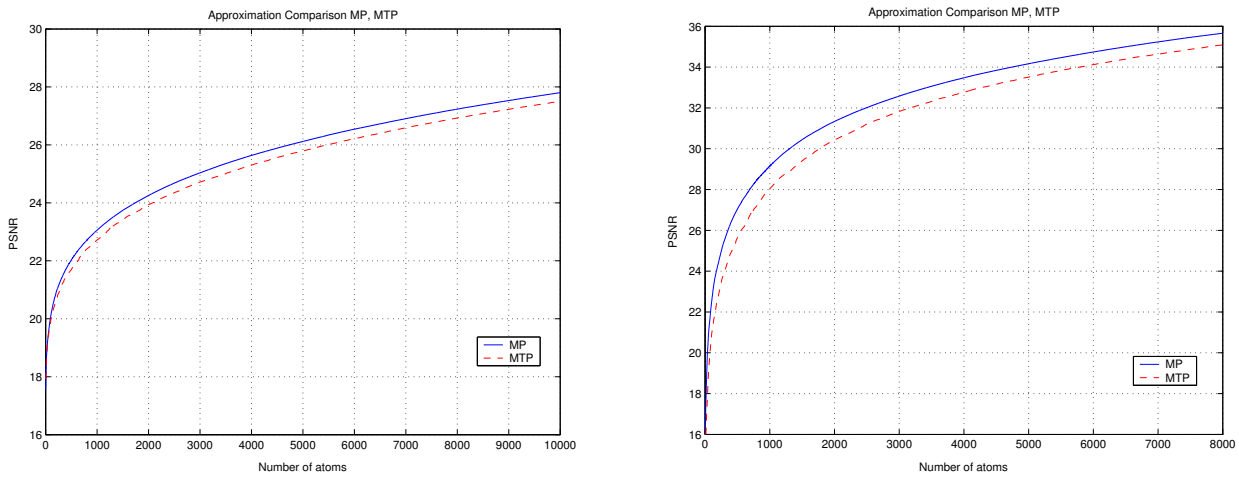
It then computes the motion mapping $\mathcal{W}$ using the estimated motion fields through the optical flow algorithm as described in [14], and a dictionary partition $\mathcal{P}_{\Omega(t_0)}(\mathcal{D})$ consisting of L ($L = 64$) sub-dictionaries. Then the M-Term Pursuit algorithm is iteratively applied on the signal $I(x, y, t)$ to find a M-term approximation of $I(x, y, t)$in M spatio-temporal atoms mapped on motion trajectory, denoted as $\mathcal{W}(g_\gamma(x_0, y_0, t_0))$, by fixing the threshold $\gamma = 0.5$ and a cumulative coherence not exceeding $\mu_s = 0.01$, or $(\mu_1(m)/m) \leq \mu_s$.

Algorithm 3 The MTP-based Decomposition Algorithm.

1: Let $I(x, y, t), t = 1..N_{GOP}$ be a block of frames, fix γ and μ_s.
2: Select a reference frame t_0 with the largest energy.
3: Find a dictionary partition $\mathcal{P}_{\Omega(t_0)}(\mathcal{D})$ with L sub-dictionaries.
4: Compute the motion mapping $\mathcal{W}$.
5: Find the collection of atoms Φ_Λ for a given γ using condition in Eq. 14
6: Discard $\phi \in \Phi_\Lambda$ from Φ_Λ if its coherence $(\mu_1(m)/m) > \mu_s$ in Φ_Λ.
7: Find the orthogonal projection of $I(x, y, t)$ on the updated Φ_Λ, i.e. $\mathcal{P}^{\perp}_{\Phi_\Lambda}(I(x, y, t))$.
8: Update the residual signal and start over from step 2.

D. Approximation Performance

The MTP algorithm has been applied on the spatio-temporal dictionary $\mathcal{D}$ with the partition $\mathcal{P}_\Omega(\mathcal{D})$ as described in the subsection VIII-B. The threshold $\gamma = 0.5$ and the coherence threshold μs is fixed to be $\mu_s = 0.01$. In all our experiments the dictionary $\mathcal{D}$ is divided into $L = 64$ sub-dictionaries $\mathcal{D}_i$ as in section VIII-B. Figure 9 shows the evolution of the error of MTP, measured in terms of the PSNR, versus the number of iterations compared against the matching pursuit algorithm for the video sequences football and foreman in CIF format, and with a group of frames (GOP = 16).



(a) Football SIF

(b) Foreman CIF

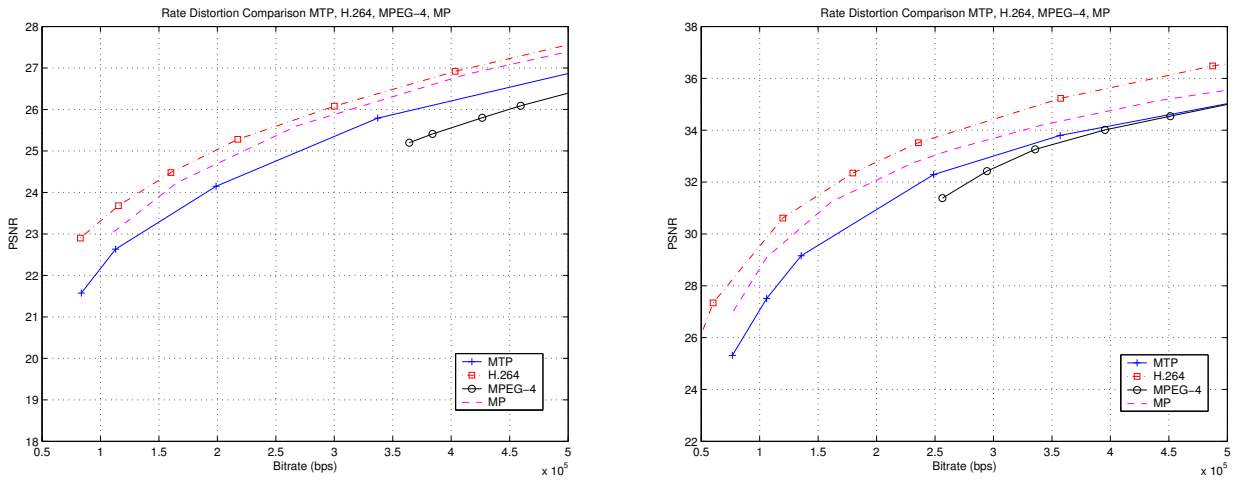
Fig. 9. Approximation comparison between MP and MTP (μ, γ) for Football and Foreman sequences

E. Progressive Coding

An embedded quantization and coding of the atoms and their coefficients based on the subset approach [14] has been applied whereas for the motion vectors, they are coded using a DPCM and an adaptive arithmetic coding.

F. Experimental Results

In this subsection, we evaluate the rate-distortion performances of the MTP-based compression scheme with the state-of-the-art video compression standards H.264 [17] and MPEG-4 [18] and our previous pure MP-based codec (the MP3D scheme), for the standard test sequences Football (SIF format) and Foreman (CIF format). It can be seen on Figure 10 that the PSNR of MTP-based codec is higher than that of MPEG-4 over all the range of low and medium bit rates for both sequences. However, it is less than that of H.264 by about 1db over the entire range under study. And finally, it is noteworthy that it performs slightly lower than the MP3D scheme, howver it has the advantage of a reduced computational complexity by an order of magnitude.



(a) Football SIF

(b) Foreman CIF

Fig. 10. Rate distortion comparison between MTP-based codec MP3D, H.264 and MPEG-4 for Football and Foreman sequences

(a) Original Frame

(b) MTP-based Codec

Fig. 11. Visual quality of the frame 1 of the Foreman CIF decoded at 580kbps, PSNR=35.76

X. Discussions and Conclusions

We have presented an adaptive approximation algorithm, the M-Term Pursuit, able to provide a sparse representations for signals in any dictionary. The approximation efficiency and the convergence of the algorithm is compared against the matching pursuit algorithm both in theory and practice. Its main strength is its improved computational complexity. In fact, the computational complexity of the MTP algorithm is reduced by at least an order of magnitude than that of MP, depending on the dictionary partitioning, the coherence, and the thresholding parameter. Once a signal decomposition is obtained, the properties of the signal components can be analyzed and processed through the structure of the selected atoms.

Two applications have been presented to demonstrate the capabilities of the MTP algorithm in compact representation and progressive coding in the field of multimedia compression and transmission: (i) image coding and (ii) video coding. The image coding scheme uses the MTP as a decomposition method in redundant visual dictionaries, or a rich collection of visual primitives able to capture image features. Once the image representation is obtained in terms of a finite sum of visual atoms, they are further quantized and coded in a progressive fashion based on

our subset approach. The performances of this scheme are favorably compared to the state-of-the-art image coders such as JPEG-2000 in terms of rate-distortion. Moreover, it provides a very flexible bit-stream able to be decoded at different spatial resolutions intrinsically.

The second application is the construction of a progressive video compression scheme. A redundant spatio-temporal dictionary is used in combination with the MTP algorithm to find an adaptive video representation in a small number of atoms compared to the size of the signal. Then, these atoms are progressively coded and transmitted using an embedded quantization and coding scheme along with the motion information. The rate-distortion behavior of the MTP-based codec is comparable to the state-of-the-art video coders such as H.264, MPEG-4 or our previously introduced MP3D scheme.

Though in our analysis no a priori information is used about the signal space, a future direction would be to assume a given model associated with a class of signals, and to adapt the dictionary to classes of signals. Another horizon will be to investigate other types of dictionary partitioning, rather than the one used in this paper which is (space or region)-based partitioning, such as scale-based or phase-based partitioning. A more compelling issue will be to partition the dictionary according classes of signals.

REFERENCES

[1] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, December 1993.

[2] Davis G., Mallat S. and Avellaneda M., "Adaptative greedy approximations," *Constructive Approximations*, 1997, springer-Verlag New York Inc.

[3] X. Huo, "Sparse image representation via combined transforms," Technical Report 1999-18, Department of Statistics Stanford University, August 1999.

[4] J. A. Tropp, "Just relax: Convex programming methods for subset selection and sparse approximation," UT-Austin, ICES 04-04, February 2004.

[5] ——, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inform. Theory*, vol. 50, no. 10, pp. 2231–2242, October 2004.

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

[7] D. L. Donoho and M. Elad, "Maximal sparsity representation via l_1 minimization," *Proc. Nat. Aca. Sci*, vol. 100, pp. 2197–2202, March Vol. 100, pp. 2197-2202, March 2003.

[8] R. A. Horn and C. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.

[9] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed. Johns Hopkins University Press, 1996.

[10] E. J. Candes and D. L. Donoho, "Curvelets- a surprisingly effective nonadaptive representation for objects with edges," in *Curve and surface fitting*, A. C. C. Rabut and L. L. Schumaker, Eds. Saint-Malo: Vanderbilt University Press, 1999.

[11] E. Le Pennec and S. Mallat, "Sparse geometric image representation with bandelets," *Submitted to IEEE Transactions on image processing*, 2003.

[12] M. N. Do and M. Vetterli, "The contourlet transform:an efficient directional multiresolution image representation," *Submitted to IEEE Transactions on image processing*, 2003.

[13] P. Frossard, P. Vandergheynst, and R. Figueras i Ventura, "High flexibility scalable image coding," in *Proc. SPIE VCIP*, Lugano (Switzerland), 2003.

[14] A. Rahmoune, P. Vandergheynst, and P. Frossard, "Flexible motion-adaptive video coding with redundant expansions," *IEEE Transactions on Circuit and Systems for Video Technology*, 2004, submitted.

[15] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Comm. ACM*, vol. 30, pp. 520–540, June 1987.

[16] I. Schoenberg, "Spline functions and the problem of graduation," *Proc. Nat. Acad. Sci.*, vol. 52, pp. 947–950, 1964.

[17] "H.264/AVC reference software, http://bs.hhi.de/ suehring/tml/."

[18] "MPEG-4 reference software, http://megaera.ee.nctu.edu.tw/mpeg/."