Sparse Approximation Using M-Term Pursuit and Application in Image and Video Coding

Adel Rahmoune, Pierre Vandergheynst, Pascal Frossard
adel.rahmoune@gmail.com {pierre.vandergheynst, pascal.frossard}@epfl.ch
Signal Processing Laboratory (LTS2/LTS4)
École Polytechnique Fédérale de Lausanne EPFL
CH- 1015 Lausanne, Switzerland

***Abstract*—This paper introduces a novel algorithm for sparse approximation in redundant dictionaries, called the M-Term Pursuit (MTP). This algorithm decomposes a signal into a linear combination of atoms that are selected in order to represent the main signal components. The MTP algorithm provides an adaptive representation for signals in any complete dictionary. The basic idea behind the MTP is to partition the dictionary into L quasi-disjoint sub-dictionaries. A k-term signal approximation is then computed iteratively, where each iteration leads to the selection of $M \leq L$ atoms based on thresholding. The MTP algorithm is shown to achieve competitive performance with the Matching Pursuit (MP) algorithm that greedily selects atoms one by one. This is due to the efficient partitioning of the dictionary. At the same time, the computational complexity is dramatically reduced compared to Matching Pursuit due to the batch selection of atoms. We finally illustrate the performance of MTP in image and video compression applications, where we show that the suboptimal atom selection of MTP is largely compensated by the reduction in complexity compared to MP.**

***Index Terms*—Sparse approximation, M-Term Pursuit, Matching pursuit, Dictionary partition, scalable image compression, scalable video compression.**

I. Introduction

This paper discusses the *sparse* signal approximation problem in redundant dictionaries, which consists in approximating signals using linear combinations of a small number of waveforms or *atoms* in a redundant *dictionary*. The objective is to compute an adaptive decomposition of a signal by expanding it into a sum of waveforms that closely match the signal structures. Such a representation provides an explanation of the signal with a series of meaningful components that describe for instance time-frequency patterns and scale-space or phase-space features. Adaptive representations are very important in signal processing applications such as audio, image or video signal analysis or compression.

Depending on the target application, the dictionary used for signal expansion can possibly be a large and redundant collection of functions. It was shown that in a finite dimensional space, computing the optimal solution for the sparse approximation of any signal in a highly redundant dictionary is an *NP-complete* problem [1]. This motivates the use of the sub-optimal greedy algorithms for computing sparse approximations. Two popular algorithms that belong to the class of the greedy approaches are represented by the Matching Pursuit (MP) algorithm [2] and its orthogonalized version (OMP) [1], [3]. MP has been proposed for computing signal expansions by iteratively selecting dictionary atoms that best correlate with the signal structures. OMP is able to improve the approximation efficiency of MP, at the expense of larger computational complexity due to an explicit orthogonalization step. Both MP and OMP have been proved to be efficient and flexible algorithms for computing sparse approximations in any type of dictionaries. One of their main drawbacks lies in the large computational complexity, since they both greedily selects only *one* atom at a time in the signal decomposition, which leads to numerous correlation computations between the residual signal and every atom in the dictionary.

In this paper, we propose to reduce the computational complexity of iterative sparse approximation algorithms by the joint selection of multiple atoms at each step of the decomposition. We build on [4] and present a new greedy algorithm called the M-Term Pursuit algorithm (MTP), which iteratively selects sets of M atoms that best approximate the signal. The atoms are picked in different quasi-incoherent blocks of a dictionary that has been partitioned a priori. The set of M atoms are selected with a simple thresholding criterion, which reports the high correlation between atoms and the residual signal. The atom selection process is followed by an orthogonal projection of the signal onto the span formed by the selected atoms; it permits to remove the contribution of the selected atoms and to update the residual signal for the next iterations of MTP. We analyze the behavior of MTP in terms of approximation performance and computational complexity. We show that the accuracy of the MTP signal decomposition stays competitive with MP due to the efficient partitioning of the dictionary while the computational complexity is reduced by a factor M. Finally, we discuss the application of MTP in image and video coding algorithms. We show that the compressed sparse approximations lead to a rate-distortion performance that is competitive with MP solutions with quite lower computational complexity. Reasonable computational complexity and good approximation performance coupled with flexible decompositions position MTP as an efficient solution for dimensionality reduction in signal processing, analysis or coding applications.

Few works have addressed the design of constructive solutions for sparse signal approximations with redundant dictio-

This work has been partly supported by the Swiss National Science Foundation under grants 2000-067859.02 and PP002-68737.

naries of atoms. In our previous work [4], a greedy algorithm has been proposed in order to have a fast image decomposition with partitioned dictionaries of atoms. We extend this work here with a detailed performance analysis and the application of the M-term pursuit to video coding applications. The reduction in complexity of the signal approximation is achieved in [5], where the dictionary is organized in a tree structure by considering the correlation between atoms. The Tree-Based Pursuit (TBP) then greedily constructs the image expansion by searching only the branches of the tree, instead of browsing the full dictionary. However, it only picks one atom at a time, while the MTP algorithm is able to select multiple uncorrelated atoms simultaneously. Other algorithms based on the orthogonal matching pursuit, such as StOMP [6] and StWOMP [7], have been also introduced to reduce the computational complexity of sparse approximations, but they do not consider the partitioning of the dictionary. The joint selection of multiple atoms might thus be suboptimal as multiple correlated atoms might be selected jointly. In a different framework where the signals are given in a compressed form, we could mention a few greedy approximation algorithms such as Compressive Sampling Matching Pursuit (CoSaMP) [8], the Subspace-Pursuit (SP) [9], and the Iterative Hard Thresholding (IHT) [10], which all try to build sparse expansions with reasonable computational complexity. In this work, we combine the selection of multiple atoms with a proper arrangement of the dictionary into incoherent partitions so that the approximation performance stays effective.

This paper is organized as follows: Section II reviews some basic concepts in matrix algebra that are extensively used throughout the paper. The MTP algorithm is described in details in Section III. Comparisons of the approximation performance and the computational complexity of MTP against MP are provided in Section IV. An application of MTP to image representation is described in Section V. Another application for progressive and scalable video coding using the MTP algorithm is discussed in Section VI.

II. Preliminaries

A. Sparse Signal Approximation

We first review the princples of sparse signal approximations with dictionaries of atoms. Let $\mathbf{f}$ be a signal that belongs to the finite dimensional complex space $\mathbb{C}^d$ or to the infinite Hilbert space $\mathcal{H}$. Both spaces are endowed with the inner product and the norm operators, which are denoted as $\langle .,. \rangle$ and $\|.\|_2$ respectively. The aim of sparse approximations is to expand $\mathbf{f}$ into a linear combination of atoms, which are denoted as g_ω. The atom index ω belongs to a large set of indexes, denoted by Ω. These atoms are drawn from a large collection of functions, which is called the dictionary; it is denoted by $\mathcal{D}$, i.e.,

$$\mathcal{D} = \{g_\omega : \omega \in \Omega\}.$$

The atoms g_ω are normalized, i.e., $\|g_\omega\|_2 = 1$. Most often, the dictionary $\mathcal{D}$ is structured in such a way that its elements $\{g_\omega\}$ are obtained through applying meaningful operators on some mother functions. The smallest possible dictionary forms a basis but dictionaries in practice are usually very redundant. The redundancy gives an increased degree of freedom in constructing signal expansions. This freedom is used in order to obtain improved approximation properties. For instance, one can form a Gabor dictionary [11], [12] by scaling, translating and modulating a Gaussian function. Generally speaking, a dictionary of time-frequency atoms $\mathcal{D} = \{g_\omega(t) : \omega \in \Omega\}$ is a very redundant set of functions in $L^2(\mathbb{R})$. It includes windowed Fourier frames and wavelet frames [13]. However, instead of expanding the signal on such a frame that is chosen a priori, the objective is to choose within $\mathcal{D}$ the time-frequency atoms that are best adapted to expand a given signal $\mathbf{f}$.

In the finite dimensional space $\mathbb{C}^d$, we represent the dictionary by a matrix $\mathbf{\Phi}$ of size $d \times N$, where N is the number of atoms in $\mathcal{D}$. The columns of $\mathbf{\Phi}$ correspond to the atoms and their order is irrelevant in the matrix $\mathbf{\Phi}$. Strictly speaking, $\mathbf{\Phi}$ is the synthesis matrix of the dictionary. Its analysis matrix is merely the conjugate transpose $\mathbf{\Phi}^*$. For example, if $\mathbf{c}$ is a coefficient vector, then $\mathbf{\Phi c}$ stands for the reconstructed signal and the quantity $\mathbf{\Phi}^*\mathbf{f}$ is the vector of the inner products between the atoms and the signal $\mathbf{f}$. A sub-dictionary $\mathcal{D}_i$ is defined as a set of atoms drawn from the dictionary $\mathcal{D}$. The atoms in $\mathcal{D}$ are indexed by the set Λ_D of cardinality N, whereas the atoms in $\mathcal{D}_i$ are indexed by Λ_i having a cardinality of N_i. In the case of finite dimensional complex space $\mathbb{C}^d$, these atoms define a synthesis matrix $\mathbf{\Phi}_i$ of size $d \times N_i$. Its corresponding analysis matrix is given by $\mathbf{\Phi}_i^*$. Both matrices $\mathbf{\Phi}_i$ and $\mathbf{\Phi}_i^*$ are sub-matrices of $\mathbf{\Phi}$ and $\mathbf{\Phi}^*$ respectively.

The objective of the k-term approximation, also referred to as *sparsity-constrained approximation*, is to provide the best approximation of a signal $\mathbf{f}$ using a linear combination of k (or fewer) atoms from the dictionary. Formally, it corresponds to solving the following problem:

$$\min_{\mathbf{c} \in \mathbb{C}^N} \|\mathbf{f} - \mathbf{\Phi c}\|_2 \qquad \text{Subject to} \qquad \|\mathbf{c}\|_0 \leq k. \quad (1)$$

B. Properties of redundant expansions

An important parameter that characterizes dictionaries of atoms is the *coherence*. It is defined as the maximum inner product magnitude between any two different atoms in the dictionary $\mathcal{D}$, i.e.,

$$\mu \stackrel{def}{=} \max_{\lambda \neq \omega} |\langle \phi_\omega, \phi_\lambda \rangle|. \quad (2)$$

When the coherence is large, the atoms are very correlated to each other; if $\mu = 1$, it means that $\mathcal{D}$ contains at least two identical atoms, or it contains two atoms, ϕ_λ and ϕ_ω, such that $\phi_\lambda = -\phi_\omega$. 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. A more accurate estimate of the dictionary structure is the *cumulative coherence* [14], [15]. It measures how much an atom can be correlated to a set of atoms. It is defined as:

$$\mu_1(m) \stackrel{def}{=} \max_{|\Lambda| = m} \quad \max_{\omega \notin \Lambda} \sum_{\lambda \in \Lambda} |\langle \phi_\omega, \phi_\lambda \rangle|. \quad (3)$$

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

Property 1 (The cumulative coherence): 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.

Another quantity measure related to the dictionary is the restricted minimum $(2, \infty)$ of a matrix, whose columns are normalized. It is defined as the following:

$$\beta \stackrel{def}{=} \min_{\mathbf{x} \neq 0} \frac{\|\mathbf{A}^* \mathbf{x}\|_\infty}{\|\mathbf{x}\|_2}. \tag{4}$$

Property 2 (The restricted minimum): If $\mathbf{A}$ spans the finite dimensional space $\mathbb{C}^d$ then β is strictly positive and less than one. This quantity, which was introduced in [2] to prove the convergence of MP in finite dimensional spaces provides a pessimistic bound on the approximation performance.

Property 3 (The singular values of the Gram matrix): Another property is that the singular values of the Gram matrix are bounded. Suppose that $\mathbf{\Phi}_\Lambda$, with $|\Lambda| = m$, is a set of some m selected atoms that are linearly independent. Then, the Gram matrix is given by $\mathbf{G} = \mathbf{\Phi}_\Lambda^* \mathbf{\Phi}_\Lambda$, whose entries are defined as $\mathbf{G}(\lambda, \omega) = \langle \phi_\lambda, \phi_\omega \rangle$.

Each singular value of $\mathbf{\Phi}_\Lambda$, $\sigma^2(\mathbf{\Phi}_\Lambda)$, satisfies the following bounds [14], [16]

$$1 - \mu_1(m-1) \leq \sigma^2(\mathbf{\Phi}_\Lambda) \leq 1 + \mu_1(m-1). \tag{5}$$

Proof: The Geršgorin Disc Theorem [17] states that each eigenvalue of $\mathbf{G}$, i.e., $\sigma^2(\mathbf{\Phi}_\Lambda)$, lies in one of the m discs that are defined by:

$$\Delta_\lambda \stackrel{def}{=} \{z : |G(\lambda, \lambda) - z| \leq \sum_{\omega \neq \lambda} |G(\lambda, \omega)|\}. \tag{6}$$

Since the atoms are normalized, it follows that $\mathbf{G}(\lambda, \lambda) = 1$. The sum in Eq. (6) is always bounded by the quantity $\mu_1(m-1)$. Thus, we have

$$|1 - \sigma^2(\mathbf{\Phi}_\Lambda)| \leq \mu_1(m-1). \tag{7}$$

■

The inverse of the Gram matrix is given by $\mathbf{G}^{-1} = \mathbf{\Phi}_\Lambda^\dagger (\mathbf{\Phi}_\Lambda^\dagger)^*$, where $\mathbf{\Phi}_\Lambda^\dagger$ is the *Moore-Penrose generalized inverse*.

Property 4 (The singular values of the generalized inverse): The singular values $\sigma^2((\mathbf{\Phi}_\Lambda^\dagger)^*)$ of the generalized inverse $\mathbf{\Phi}_\Lambda^\dagger$ are also bounded as follows:

$$\frac{1}{1 + \mu_1(m-1)} \leq \sigma^2((\mathbf{\Phi}_\Lambda^\dagger)^*). \tag{8}$$

Proof: The eigenvalues of $\mathbf{G}^{-1}$, which are also the singular values $\sigma^2((\mathbf{\Phi}_\Lambda^\dagger)^*)$, correspond to the inverse of the eigenvalues of the Gram matrix $\mathbf{G}$. Hence, we can use the upper bound for $\sigma^2(\mathbf{\Phi}_\Lambda)$ in Eq. (5) to obtain the lower bound for $\sigma^2((\mathbf{\Phi}_\Lambda^\dagger)^*)$ as it is given in Eq. (8). ■

III. The M-Term Pursuit Algorithm

We present now in details the MTP algorithm that is an iterative sparse decomposition algorithm based on the joint selection of sets of M atoms. It relies on a partitioning of the dictionary into quasi-incoherent sub-dictionaries, so that the approximation performance is penalized only marginally by the joint selection of sets of atoms. The computational complexity is further reduced by an order M compared to a classical Matching Pursuit algorithm.

A. Dictionary Partitioning

A dictionary partition $\mathcal{P}(\mathcal{D})$ is defined as a collection of L disjoint sub-dictionaries, $\mathcal{D}_i \subseteq \mathcal{D}$, in such a way that their union preserves the dictionary $\mathcal{D}$ integrity. An assumption is made on the number of sub-dictionaries L, which should be less than or equal to d. Formally, it is defined as:

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

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

In matrix notation, the concatenation of the sub-matrices $\mathbf{\Phi}_i$ in a row-wise manner must reproduce the original matrix $\mathbf{\Phi}$, regardless of the order of the columns. Of course, one should expect that a sub-dictionary $\mathcal{D}_i$ carries some meaningful information about the signal. Two special cases are predominant for dictionary analysis. One case is when each $\mathcal{D}_i$ is an orthonormal basis. In such a case, the dictionary $\mathcal{D}$ is simply the union of L orthonormal bases (e.g., the union of a wavelet and a Fourier basis). This class of structured dictionaries has received lot of research efforts and led to very interesting results [18]–[20]. The other case, which is of significant importance too, happens when $\mathcal{D}_i$ has some given coherence[1] between its own atoms, but it has *very small* coherence (i.e., $\mu \leq 0.25$) with atoms that belong to distinct sub-dictionaries, i.e., these atoms are *almost independent* or incoherent. Thus, we call this type of dictionary partition an *incoherent partition*. Though there is very limited work in this direction, we think that this type of partition is as important as the previous case. In incoherent partitions, if M atoms are selected such that each atom pertains to a different sub-dictionary $\mathcal{D}_i$, then it is likely that these atoms are linearly independent and their cumulative coherence μ_1 shall be *small*. Of course, the number M of these atoms should satisfy $M \leq L$.

B. Signal approximation

The MTP algorithm can be regarded as a constructive approach for solving the sparse approximation problem in Eq. (1). It tries to find the best k-term approximation - in a ℓ_2 sense - of the signal $\mathbf{f}$. In other words, the problem can be formulated as finding

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

$$\tilde{\mathbf{f}} = \sum_{i=0}^{k-1} c_i g_{\gamma_i} \quad \text{such that} \qquad \|\mathbf{f} - \tilde{\mathbf{f}}\|_2 \leq \epsilon, \tag{9}$$

where g_{γ_i} are atoms in $\mathcal{D}$ and c_i are coefficients. MTP is a greedy algorithm that iteratively picks M atoms at a time, in order to approximately solve the above problem with a tractable complexity.

In more details, let $\mathcal{P}(\mathcal{D})$ be an incoherent partition into L sub-dictionaries of the dictionary $\mathcal{D}$ having a synthesis matrix $\mathbf{\Phi}$. Each sub-dictionary $\mathcal{D}_i$ is associated with its synthesis matrix $\mathbf{\Phi}_i$. We suppose also that $L \leq d$. The decomposition of any signal $\mathbf{f}$ can be performed iteratively by the MTP. MTP initially finds the best M-term approximation of the signal $\mathbf{f}$, then updates it in order to form the residual signal $R^j\mathbf{f}$. This process is repeated iteratively until convergence, where each iteration comprises two operations that are respectively the atom selection and the residue computation steps.

During the atom selection step, all the inner products between each atom in $\mathcal{D}$ and $\mathbf{f}$ are calculated. The quantities $\|\mathbf{\Phi}^*\mathbf{f}\|_\infty$ and $\|\mathbf{\Phi}_i^*\mathbf{f}\|_\infty$ denote the largest inner product magnitude in $\mathcal{D}$ and $\mathcal{D}_i$ respectively. Then, only *the atom* $\phi_i \in \mathcal{D}_i$, which satisfies the thresholding condition for a fixed threshold γ is selected, i.e.,

$$\frac{\|\mathbf{\Phi}_i^*\mathbf{f}\|_\infty}{\|\mathbf{\Phi}^*\mathbf{f}\|_\infty} \geq \gamma, \tag{10}$$

for $0 \leq \gamma \leq 1$. Note that there is no guaranty that an atom is selected in each dictionary. Now, the collection of the selected atoms, which is denoted as $\mathbf{\Phi}_\Lambda$, has a cardinality of M, with $M \leq L$. Moreover, these atoms are assumed to be linearly independent since each atom lies in a different sub-dictionary. These sub-dictionaries are incoherent with each others. The second step consists in an orthogonal projection $P_V\mathbf{f}$ of $\mathbf{f}$ onto the span of the selected atoms indexed in Λ. Then, the signal is updated by removing the quantity $P_V\mathbf{f}$ from $\mathbf{f}$. The formal description of the M-term Pursuit algorithm is described in Algorithm 1.

C. Approximation rate

We analyze now the approximation performance of MTP. Before stating the main theorem, let us start with the following lemma about the existence of $\mathbf{\Phi}_\Lambda$ for a given threshold γ.

Lemma 1 (The Cardinality of $\mathbf{\Phi}_\Lambda$): Suppose that γ is used to select $\mathbf{\Phi}_\Lambda$ in a single iteration such that:

$$\mathbf{\Phi}_\Lambda = \{\phi_\lambda : \phi_\lambda = \arg \max_{\phi \in \mathbf{\Phi}_i} (|\langle \phi, \mathbf{f} \rangle|), \frac{\|\mathbf{\Phi}_i^*\mathbf{f}\|_\infty}{\|\mathbf{\Phi}^*\mathbf{f}\|_\infty} \geq \gamma, i = 1 \dots L\}. \tag{11}$$

Let the number of selected atoms be denoted by $M = |\mathbf{\Phi}_\Lambda|$. Let also β be given by Eq. (4) for the matrix $\mathbf{\Phi}$. Then, for a given thresholding condition, the quantity M is upper-bounded as follows:

$$M < \frac{1}{\gamma^2}\left(\frac{\mu_1(M-1)}{\beta^2} + \frac{1}{\beta^2} - 1\right) + 1. \tag{12}$$

Note finally that the cardinality has to respect $|\mathbf{\Phi}_\Lambda| \leq L$ by construction, since we pick at most one atom per sub-dictionary. Thus, the number of selected atoms cannot be larger than the number of sub-dictionaries.

Algorithm 1 The MTP Algorithm

INPUT:

The signal $\mathbf{f}$ and the number of iterations J.

OUTPUT:

The set of selected atoms $\mathcal{S}$, the approximation $\mathbf{a}_J$ and residual $R^J\mathbf{f}$.

PROCEDURE

1. Initialize the residual signal $R^0\mathbf{f} = \mathbf{f}$, the approximation $\mathbf{a}_0 = 0$, the set of atoms $\mathbf{\Phi}_\Lambda = \emptyset$, the set $\mathcal{S} = \emptyset$ and the iteration number $j = 1$.
2. For $i = 1$ to L
   Find $\phi_i = \arg_{\phi \in \mathbf{\Phi}_i}(\|\mathbf{\Phi}_i^* R^{j-1}\mathbf{f}\|_\infty)$.
   IF $\frac{|\langle R^{j-1}\mathbf{f}, \phi_i \rangle|}{\|\mathbf{\Phi}^* R^{j-1}\mathbf{f}\|_\infty} \geq \gamma$
   Then $\mathbf{\Phi}_\Lambda \leftarrow \mathbf{\Phi}_\Lambda \cup \{\phi_i\}$.
3. Determine the orthogonal projection P_V onto the span of $\mathbf{\Phi}_\Lambda$.
4. Update $\mathcal{S}$ and compute the approximation and the residual:

$$\begin{aligned} \mathcal{S} &\leftarrow \mathcal{S} \cup \Phi_\Lambda, \\ \mathbf{a}_j &\leftarrow \mathbf{a}_{j-1} + P_V R^{j-1}\mathbf{f}, \\ R^j\mathbf{f} &\leftarrow R^{j-1}\mathbf{f} - P_V R^{j-1}\mathbf{f}. \end{aligned}$$

5. Increment j, empty $\mathbf{\Phi}_\Lambda$ ($\mathbf{\Phi}_\Lambda = \emptyset$) and go to step 2 if $j \leq J$.

Proof: The best approximation of $\mathbf{f}$ on $\mathbf{\Phi}_\Lambda$ is obtained through the orthogonal projection on the span of $\mathbf{\Phi}_\Lambda$. This projection ($P_V\mathbf{f}$) is defined as:

$$P_V\mathbf{f} = \mathbf{\Phi}_\Lambda \mathbf{\Phi}_\Lambda^\dagger \mathbf{f}, \tag{13}$$

where

$$\mathbf{\Phi}_\Lambda^\dagger = (\mathbf{\Phi}_\Lambda^* \mathbf{\Phi}_\Lambda)^{-1} \mathbf{\Phi}_\Lambda^*. \tag{14}$$

Denote the vector $\mathbf{x}$ as $\mathbf{x} = \mathbf{\Phi}_\Lambda^*\mathbf{f}$ and the Gram matrix as $\mathbf{G} = \mathbf{\Phi}_\Lambda^*\mathbf{\Phi}_\Lambda$. One can easily verify that

$$\|P_V\mathbf{f}\|_2^2 = \mathbf{x}^*\mathbf{G}^{-1}\mathbf{x}, \tag{15}$$

where the inverse of the Gram matrix $\mathbf{G}^{-1}$ is also defined as $\mathbf{\Phi}_\Lambda^\dagger(\mathbf{\Phi}_\Lambda^\dagger)^*$. This matrix is positive definite, hence we can bound the approximation according to [17] as,

$$\|P_V\mathbf{f}\|_2^2 \geq \sigma_{min}^2((\mathbf{\Phi}_\Lambda^\dagger)^*)\|\mathbf{x}\|_2^2. \tag{16}$$

By using the result of Eq. (8), we get

$$\|P_V\mathbf{f}\|_2^2 \geq \frac{1}{1 + \mu_1(M-1)}\|\mathbf{x}\|_2^2. \tag{17}$$

By the thresholding condition, we know that for all admitted atoms in $\mathbf{\Phi}_\Lambda$, their corresponding inner product magnitudes satisfy $|x_i| \geq \gamma\|\mathbf{x}\|_\infty$. Thus, we have

$$\|\mathbf{x}\|_2^2 \geq (1 + (M-1)\gamma^2)\|\mathbf{x}\|_\infty^2.$$

Substituting this expression into Eq. (17) gives

$$\|P_V\mathbf{f}\|_2^2 \geq \frac{(1 + (M-1)\gamma^2)}{1 + \mu_1(M-1)}\|\mathbf{x}\|_\infty^2. \tag{18}$$

Since we have $\mathbf{x} = \mathbf{\Phi}_\Lambda^* \mathbf{f}$, we combine the bounds in Eq. (4) on $\mathbf{\Phi}_\Lambda^*$ and we use Eq. (18) in order to obtain the following results:

$$\|P_V \mathbf{f}\|_2^2 \geq \beta^2 \frac{(1+(M-1)\gamma^2)}{1+\mu_1(M-1)} \|\mathbf{f}\|_2^2. \tag{19}$$

Since $P_V \mathbf{f}$ is an orthogonal projection, it must always satisfy the following:

$$\|P_V \mathbf{f}\|_2^2 \leq \|\mathbf{f}\|_2^2.$$

Finally, by using Eq. (19) and rearranging the terms, we get the upper bound of Eq. (12). ∎

Theorem 1 (The approximation rate of MTP): Let $\mathcal{P}(\mathcal{D})$ be a partition of $\mathcal{D}$ and $\mathbf{f}$ be any signal in the Hilbert space $\mathcal{H}$. The threshold γ is selected such that

$$\gamma \leq \frac{1}{\beta} \sqrt{\frac{\mu_1(M-1)+(1-\beta^2)}{(M-1)}}. \tag{20}$$

Suppose that $\mathbf{\Phi}_\Lambda$ exists with $m = |\mathbf{\Phi}_\Lambda|$ and that the MTP returns an approximation $\mathbf{a}_M$ in a single iteration. Then the approximation error is bounded by:

$$\|\mathbf{f} - \mathbf{a}_M\|_2^2 \leq \left(1 - \frac{\beta^2(1+(M-1)\gamma^2)}{1+\mu_1(M-1)}\right) \|\mathbf{f}\|_2^2. \tag{21}$$

Proof: The m-term approximation $\mathbf{a}_M$ of $\mathbf{f}$ produces an error $\mathbf{f} - \mathbf{a}_M$. This error satisfies the triangular equality, i.e.,

$$\|\mathbf{f} - \mathbf{a}_M\|_2^2 = \|\mathbf{f}\|_2^2 - \|P_V \mathbf{f}\|_2^2, \tag{22}$$

because of the orthogonal projector.

Now, by substituting the lower bound on $\|P_V \mathbf{f}\|_2$ from Eq. (19), we get the approximation error bound as stated in Eq. (21). ∎

The previous bound holds only for a single iteration. Suppose that the same algorithm is run up to J-th iteration 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_j, 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 $\mathbf{a}_k$ with k atoms such that $k = \sum_{j=1}^{J} M_j$. Then, the approximation error $\mathbf{f} - \mathbf{a}_k$ is bounded as:

$$\|\mathbf{f} - \mathbf{a}_k\|_2^2 \leq \|\mathbf{f}\|_2^2 \prod_{j=1}^{J} \left(1 - \frac{\beta^2(1+(M_j-1)\gamma_j^2)}{1+\mu_1(M_j-1)}\right). \tag{23}$$

This corollary can be proven by induction using the results of Theorem 1.

Finally, it should be noted that the parameter γ 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 at each iteration of MTP. On the other extreme, when $\gamma = 0$, it means that all the L atoms are selected. Notice also that the above analysis does not include any indication about how to find a threshold value γ that guarantees the selection of M atoms at each iteration. While using a constant value of M at each iteration is not critical in MTP, two practical approaches can be used to determine γ:

- A top-down approach: starting from one ($\gamma = 1$ and $|\mathbf{\Phi}_\Lambda| = 1$), the parameter γ is slowly decreased until $|\mathbf{\Phi}_\Lambda|$ reaches M.
- A bottom-up approach: starting from zero ($\gamma = 0$ and $|\mathbf{\Phi}_\Lambda| = L$), the parameter γ is smoothly increased until $|\mathbf{\Phi}_\Lambda|$ reaches M.

IV. Performance Comparisons with MP

In this section, we discuss the implications of the error bound in Theorem 1 and Corollary 1 for sparse approximations. The effect of the coherence μ_1, the dictionary parameter β and the threshold γ are also discussed along with a comparison against the original Matching Pursuit (MP) algorithm.

A. Approximation performance

The error bound in Eq. (21) 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 $\mu_1(m) \leq m\mu$ and γ_j equals to γ for all iterations, the approximation error can be further bounded as:

$$\|\mathbf{f} - \mathbf{a}_M\|_2^2 \leq \left(1 - \frac{\beta^2(1+(M-1)\gamma^2)}{1+(M-1)\mu}\right) \|\mathbf{f}\|_2^2. \tag{24}$$

Let $h_1(M, \mu, \beta)$ be the decay factor in Eq. (24) defined by

$$h_1(M, \mu, \beta) = \left(1 - \frac{\beta^2(1+(M-1)\gamma^2)}{1+(M-1)\mu}\right). \tag{25}$$

Finally, we denote by $h_2(M, \mu, \beta)$ the decay factor of the iterative MTP, which is written as

$$h_2(M, \mu, \beta) = \prod_{j=1}^{J} \left(1 - \frac{\beta^2(1+(M_j-1)\gamma^2)}{1+(M_j-1)\mu}\right). \tag{26}$$

Recall that the MP algorithm [2] has an error bound of the M-term approximation given by

$$\|\mathbf{f} - \mathbf{a}_M\|_2^2 \leq \left(1 - \beta^2\alpha^2\right)^M \|\mathbf{f}\|_2^2. \tag{27}$$

Without the loss of generality, we assume that $\alpha = 1$ (i.e., MP selects iteratively the *best* atom). Let us define the decay rate of MP as $h_{mp}(M, \beta)$ written as:

$$h_{mp}(M, \beta) = \left(1 - \beta^2\right)^M. \tag{28}$$

We now compare experimentally the decay rates of MTP and MP that have been computed above. In the following experiments, the threshold γ in MTP is chosen in the interval [0.4, 0.7]. In general, small values of γ accounts for some inefficiency in the approximation performance, whereas values close to one lead MTP to perform close to MP, i.e., it picks 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. (12). Similarly, some sample values for the parameter β have been chosen in order to characterize different classes of dictionaries (e.g., 0.3, 0.4, 0.5, and 0.6). Different values are assigned to the coherence μ in the set of the selected atoms $\mathbf{\Phi}_\Lambda$ (0.05, 0.02, 0.01 and 10^{-6}), these values can indicate the global trend of the approximation error and they are independent from each other. With no surprise,

one should expect that the smaller is the coherence μ *in the set* $\mathbf{\Phi}_\Lambda$, the better must be the approximation in general. Of course, this property is in harmony with the fact that, in general, the approximation is better in a redundant dictionary $\mathcal{D}$ that is characterized by a quite large coherence μ.

We first compare the evolution of the decay rate given by h_1 and h_{mp} in MTP and MP respectively. The error bounds for the MTP and MP have been plotted for different values of β and μ versus M according to the parameter settings, described above. The graphs of h_1 and h_{mp} are displayed in Figures 1, 2 and 3(a) with the following observations:

- In Figure 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.
- In Figure 2, we see 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.
- In Figure 3(a), the effect of the β parameter is shown when fixing the values of μ and γ. We see clearly that the decay rate of the function h_1 is related to β. For instance, when β increases, the decay rate increases also and vice versa.

Then we study the behavior of the rates given by h_2 and h_{mp}. The error bound function h_2 of the algorithm described in Corollary 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 selected atoms equals to $L = |\mathbf{\Phi}_\Lambda| = 8$ per iteration. We see clearly some jumps occurring between iterations. These discontinuities are due to the fact that h_1 has a fast decay rate with few atoms then it tends to saturate. Such saturations are avoided by going through more iterations. So in order to achieve a k-term approximation in any dictionary, a compromise should be reached between the number of iterations J and the size of $\mathbf{\Phi}_\Lambda$ in each iteration.

It is very important to note that all the previous analysis on the error bound is performed on the theoretical worst-case scenario of $\mu_1(M)$, which would reach $M \cdot \mu$. This is a very rough bound on μ_1. In practice, the approximation performances of the MTP algorithm are very close to those related to MP, as we will be seen later.

B. Computational Complexity

We discuss now the computation complexity of MTP. The reduced computational complexity is the main strength of MTP compared to the MP algorithm, though the actual complexity reduction is thoroughly dependent on the dictionary. In one iteration, the MTP performs N inner products between each atom in $\mathcal{D}$ and the residual function $R^j\mathbf{f}$. Assuming that each inner product requires p operations, the algorithm accumulates an order of $O(N \cdot p)$ operations. Once the M atoms are selected, it needs to solve a linear system. By using a Conjugate Gradient [21] solver, it requires a complexity of $O(q \cdot M)$, where q is a constant. Thus, for the M-term approximation in a given iteration, the MTP culminates an order of computational complexity of:

$$O(N \cdot p + M \cdot q). \tag{29}$$

The matching pursuit, however, has a computational complexity of order $O(N \cdot M \cdot p)$ for obtaining an approximation with same number of atoms M.

In very redundant dictionaries, i.e., when the number of atoms $N \gg M$, the second term in Eq. (29) is almost negligible compared to the first term. Thus asymptotically, the MTP can achieve a speed-up gain of up to M times when compared to MP.

In the next sections we discussed two applications of MTP in image and video compression using very redundant dictionaries.

V. Application of MTP to Image Representation

We illustrate in this section the use of MTP for image representation with a collection of image primitives, which build up the dictionary $\mathcal{D}_s$. Once the image representation is computed, the selected atoms are encoded in a compressed bitstream.

A. Image Dictionary

The dictionary $\mathcal{D}_s$ is constructed using the same methodology as proposed in [22]. It is generated by applying affine transformations on two mother functions, which are a 2-D Gaussian $g_1(\mathbf{x}) = \frac{1}{\sqrt{\pi}} e^{-(x_1^2+x_2^2)}$ and its 2^{nd} partial derivative (a ridge-like function) $g_2(\mathbf{x}) = \frac{2}{\sqrt{3\pi}}(4x_1^2 - 2)e^{-(x_1^2+x_2^2)}$. The 2-D Gaussian is used in order to extract the low frequency components. Its 2^{nd} partial derivative is used to capture image singularities like edges and contours. The affine operator is a composition of translation, scaling and rotation of the mother functions as follows:

$$\mathcal{U}_{(\mathbf{b},\mathbf{a},\theta)}g = \frac{1}{\sqrt{a_1 a_2}} g(r_{-\theta}(\frac{x_1 - b_1}{a_1}, \frac{x_2 - b_2}{a_2}))\,, \tag{30}$$

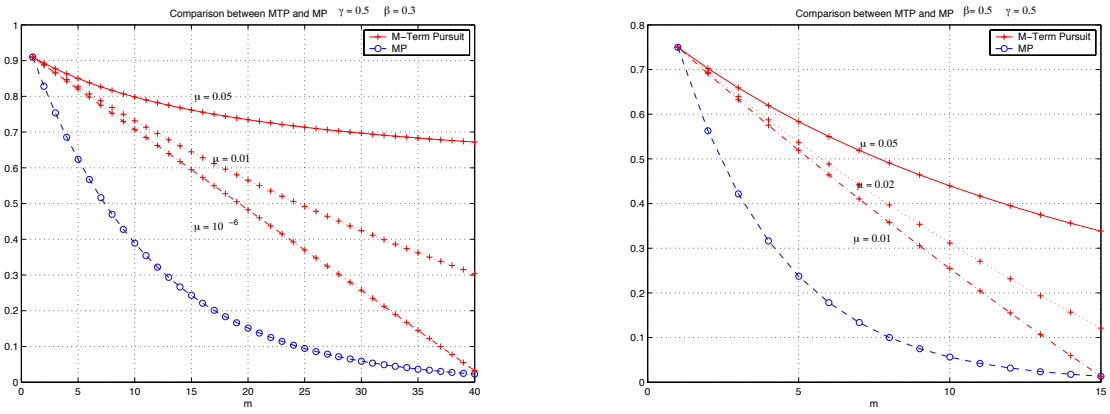
where $r_{-\theta}$ is a rotation matrix of angle θ, $\mathbf{b}$ is the translation and $\mathbf{a}$ is the scale. Each atom $g_\omega(\mathbf{x}) \in \mathcal{D}_s$ is uniquely identified by the tuple $\omega = (\nu, \mathbf{b}, \mathbf{a}, \theta)$ that forms the atom index.

B. Dictionary Partitioning

The dictionary $\mathcal{D}_s$ is partitioned spatially into L sub-dictionaries $\mathcal{D}_i$. Each sub-dictionary $\mathcal{D}_i$ consists of all functions $g_\omega(\mathbf{x})$'s whose center $\mathbf{x}_0$ belongs to a region Ω_i. In this analysis, an image $I(\mathbf{x})$ of size (X_1, X_2) is divided into L non-overlapping rectangular blocks of size $(X_{1_\Omega}, X_{2_\Omega})$. Every block is associated to a region Ω_i and thus to a sub-dictionary $\mathcal{D}_i$. It is illustrated in Figure 4.

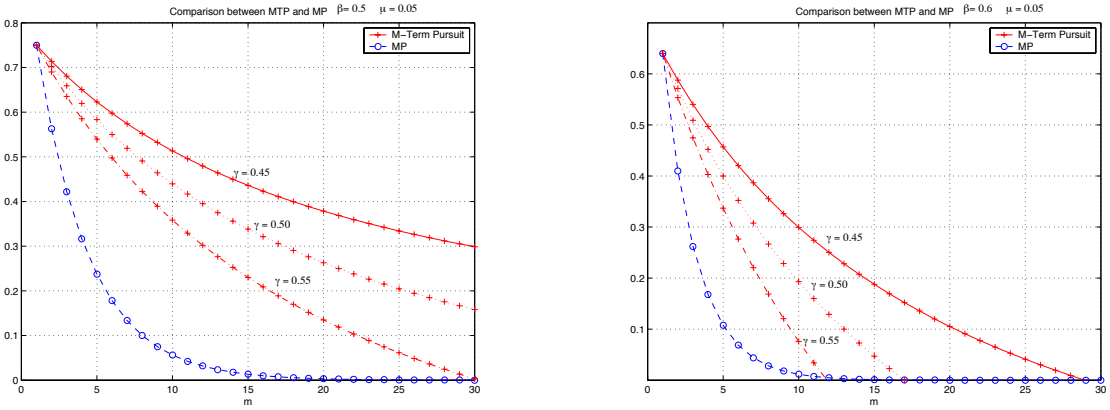
More formally, the sub-dictionary $\mathcal{D}_i$ is given as

$$\mathcal{D}_i = \{g_\omega(\mathbf{x} - \mathbf{x}_0) : \mathbf{x}_0 \in \Omega_i\},$$



(a) $\beta = 0.3$ and $\gamma = 0.5$. (b) $\beta = 0.5$ and $\gamma = 0.5$.

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



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

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

such that

$$\Omega = \bigcup_{i=1}^{L} \Omega_i,$$
$$\Omega_i \bigcap \Omega_j = \emptyset, \quad \text{for} \quad i \neq j.$$

This class of partitioning is called space partitioning and it is denoted as $\mathcal{P}_\Omega(\mathcal{D})$. It guarantees that the coherence between the atoms in different regions is likely to be small. However, the atoms that are lying near the border of the regions Ω_i's or the atoms that have large support do not meet this condition. Most often, the atoms that are located near the boundaries of a given region Ω_i have a support extending to other regions. Nevertheless, they are considered to belong to Ω_i as long as their center lies in the block.

Other dictionary partitioning approaches based on either scales or orientations, or combinations of all transformations, are also possible. Another interesting dictionary partitioning scheme would be through image segmentation, where homogeneous regions would be associated to sub-dictionaries. These partitioning schemes are however left as future work.

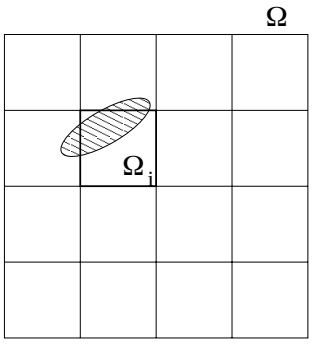
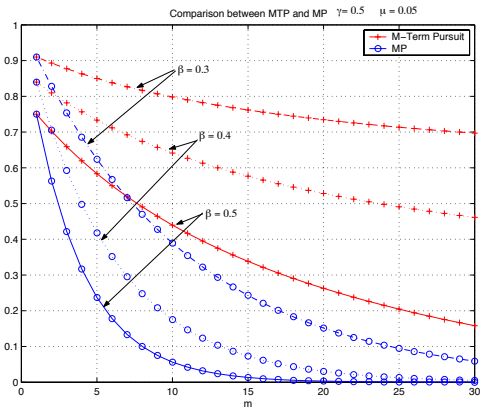


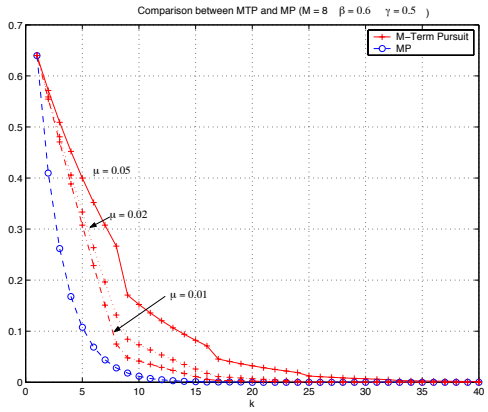
Fig. 4. An illustration of space partitioning. The image domain Ω is divided into rectangular regions. For instance, Ω_i is delimited by the bold lines. The atom g_ω that is represented by the shaded lines belongs to the region Ω_i because its center $\mathbf{x}_0$ belongs to Ω_i, even though it has a support overlapping with the neighboring regions.

C. MTP Image Decomposition

Our illustrative application of MTP to image decomposition is built as follows. The dictionary partition $\mathcal{P}_\Omega(\mathcal{D})$ is composed of L sub-dictionaries according to the previous description. The parameter L is fixed to $L = 64$. The MTP algorithm iteratively decomposes the image into a finite number of spatial



(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.

atoms. It is presented formally in Algorithm 2. Let $R^jI(\mathbf{x})$ denote the residual image at the iteration j, and by convention $R^0I(\mathbf{x}) = I(\mathbf{x})$. At iteration j, the MTP algorithm performs two steps: the selection step and the projection step.

Selection Step: During the selection step, the inner product between all the atoms in $\mathcal{D}_s$ and the residual image $R^jI(\mathbf{x})$ are calculated. Then, M_j atoms are selected. At most one atom is selected from each sub-dictionary $\mathcal{D}_i$ according to the following two criteria:

- The atom $g_{\omega_i}(\mathbf{x})$ has an inner product that satisfies the threshold γ, i.e.,

$$|\langle R^{j-1}I(\mathbf{x}), g_{\omega_i}(\mathbf{x})\rangle| \geq \gamma \sup_{\omega\in\Omega} |\langle R^{j-1}I(\mathbf{x}), g_{\omega}(\mathbf{x})\rangle|,$$

- The atom $g_{\omega_i}(\mathbf{x})$ has a coherence $\mu_1(M_j - 1)$ in the set Φ_Λ, which satisfies the coherence threshold μ_s, i.e.,

$$\frac{\mu_1(M_j - 1)}{M_j - 1} \leq \mu_s.$$

The atom admission process into the set Φ_Λ is performed following the non-decreasing order for their inner product magnitudes. This order is important so that the atoms that have large contributions in the image residual, i.e., large $|\langle R^jI, g_\omega\rangle|$, are kept in the set Φ_Λ. This operation is described in Steps 2 and 3 in Algorithm 2.

D. MTP Approximation Performance

The MTP algorithm has been applied to the dictionary $\mathcal{D}_s$ with the partition $\mathcal{P}_\Omega(\mathcal{D})$ as described in the previous section. We set the threshold γ to 0.7 and the coherence threshold μ_s is assigned values in 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 5 shows the behavior of the approximation error of MTP and MP, measured in terms of PSNR versus the number of atoms for $\gamma = 0.7$ and different μ_s's. It is clear that the gap between the approximation performance of both algorithms decreases with smaller μ_s and larger γ values because, in this case, the selected atoms are almost uncorrelated and have a large energy contribution (defined by

Algorithm 2 The MTP Image Decomposition Algorithm.

INPUT:

The image I, the number of iterations J, the threshold γ and μ_s.

OUTPUT:

The set of selected atoms $\mathcal{S}$, the approximation a_J and residual R^JI.

PROCEDURE

1. Initialize the residual signal $R^0I = I$, the approximation $a_0 = 0$, the set of atoms $\Phi_\Lambda = \emptyset$, the set $\mathcal{S} = \emptyset$ and the iteration number $j = 1$.
2. Set $M_j = L$.
3. For $i = 1$ to L
   Find $g_{\omega_i} = \arg\sup_{\omega\in\Omega_i} |\langle R^{j-1}I, g_\omega\rangle|$,
   If $|\langle R^{j-1}I, g_{\omega_i}\rangle| \geq \gamma \sup_{\omega\in\Omega} |\langle R^{j-1}I, g_\omega\rangle|$
   Then $\Phi_\Lambda \leftarrow \Phi_\Lambda \cup \{g_{\omega_i}\}$.
4. Discard g_ω and decrement M_j from Φ_Λ if its coherence $(\mu_1(M_j - 1)/(M_j - 1)) > \mu_s$ in Φ_Λ, following the non-decreasing order of $|\langle R^{j-1}I, g_\omega\rangle|$.
5. Determine the orthogonal projection $P_V R^{j-1}I$ onto the span of Φ_Λ, update $\mathcal{S}$, the approximation and the residual:

$$\begin{aligned} \mathcal{S} &\leftarrow \mathcal{S} \cup \Phi_\Lambda, \\ a_j &\leftarrow a_{j-1} + P_V R^{j-1}I, \\ R^jI &\leftarrow R^{j-1}I - P_V R^{j-1}I. \end{aligned}$$

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

γ) during a single iteration. With $\gamma = 0.7$ and $\mu_s = 0.01$, MTP performs almost as well as MP.

It has been observed that the number of selected atoms increases along the pursuit; 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 parts of the signal, which tend to have a support expanding over different regions Ω_i. Hence, these atoms with overlapping supports are characterized by a *large* coherence. Then, imposing a small coherence threshold μ_s on the atoms

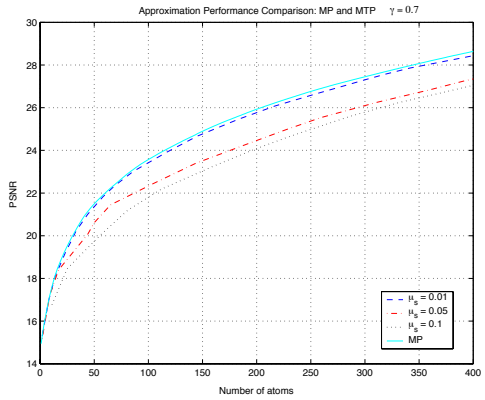


Fig. 5. Approximation performance comparison between MP and MTP for Lena image (256x256)

of Φ_Λ tends to reduce its cardinality. On the other hand, when the number of iteration increases, the number of selected atoms per iteration stabilizes; these atoms are mainly chosen to represent edges that are localized, and these atoms are therefore almost decorrelated. The restriction on their selection comes principally from the threshold γ. A direct consequence of this observation is that the complexity gain of the MTP over MP, which is defined mainly by the cardinality of Φ_Λ, becomes significant after few iterations.

Figure 6 shows the computational complexity as a function of the residual error measured in PSNR for both MTP and MP algorithms applied to the Lena image (256x256). One can see clearly that the complexity reduction in MTP becomes very significant at high PSNR values. This is due to the fact that, at this stage of the approximation, the selected atoms tend to be uncorrelated, i.e., in a single iteration the number of selected atoms by the MTP tends to the maximum L whereas MP keeps selecting only one atom per iteration. For instance, at a PSNR of 28.6 DB, MP requires almost 400 time units in contrast to the 41 units required for the MTP, thus accounting for a speed-up of 9.68.

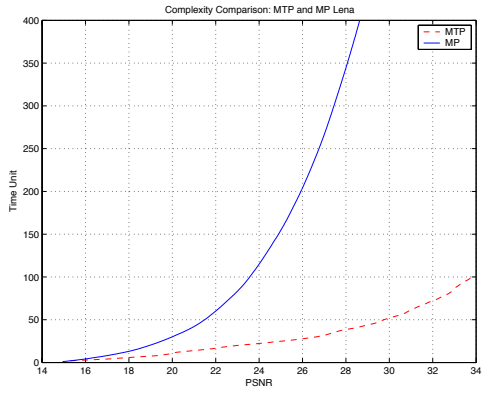
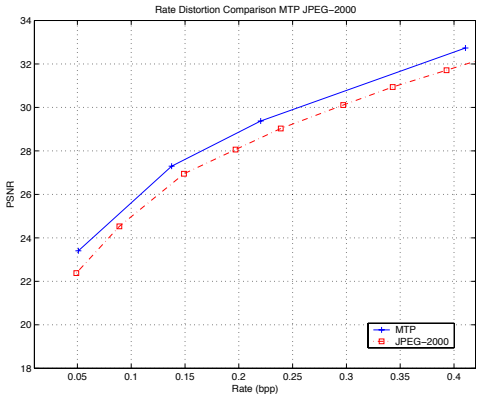


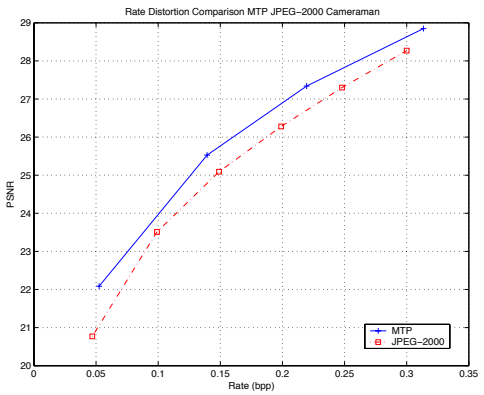
Fig. 6. Complexity comparison between MTP (with $\gamma = 0.7$ and $\mu = 0.01$) and MP algorithms: Time unit vs PSNR for Lena image.

E. Image Coding Performance

An interesting application for compact image representation is compression and especially progressive compression, which consists in embedded quantization and coding operations. This is achieved by using the subsets approach that is introduced in our previous work [23]. 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 and Cameraman images of size 256x256 are used in our experiments. 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 for both images, i.e. up to 0.4 bpp. Figure 8 show the visual quality of the MTP-based codec



(a) Lena (256x256)



(b) Cameraman (256x256)

Fig. 7. Rate distortion comparison between MTP ($\gamma = 0.7$, $\mu_s = 0.01$),and JPEG-2000.

when applied to Lena image. When decoded at a target bit rate of 0.41 bpp, its PSNR is 32.73 dB. And one can see that all the edges, contours and prominent features present in the image have been captured, which gives some smoothness to the reconstructed image. Note that we illustrate here the rate-distortion behavior in the low to medium rate range, as MP and thus MTP are mostly efficient in this region. In higher rate, the dictionary proposed here, and the greedy atom selection in MP and MTP does not permit to capture well the texture information and to compete with state-of-the-art coders [24].

VI. Application to Video Representation and Compression

Video representation is the operation of decomposing the video sequence into a linear combination of spatio-temporal

(a) Original Lena (b) MTP encoding

Fig. 8. The Lena image (256x256) encoded with the MTP sample algorithm at 0.41bpp [PSNR = 32.73dB].

waveforms, or atoms that are aligned along motion trajectories. This operation requires a decomposition algorithm over a dictionary $\mathcal{D}_{st}$ of motion-adaptive spatio-temporal atoms. The MTP algorithm is used as a decomposition algorithm, and the dictionary $\mathcal{D}_{st}$ consists of spatio-temporal atoms mapped along motion trajectories, having spatial component as described in subsection V-A and a temporal component able to capture the temporal signal evolution. A target application of scalable compression, which has gained more focus recently in ubiquitous applications, is introduced. Recall that the goal of scalable coding is to generate a single embedded bit-stream able to be decoded efficiently over a given range of bit-rate and at various resolutions.

A. Spatio-temporal Dictionary

The spatio-temporal dictionary $\mathcal{D}_{st}$ is constructed by using three-dimensional multiscale ridge-like functions $\psi_{st}(\mathbf{x},t)$ and three-dimensional smooth functions $\phi_{st}(\mathbf{x},t)$. These functions are defined as $\phi_{st}(\mathbf{x},t) = g_1(\mathbf{x})\phi_\tau(t)$ and $\psi_{st}(\mathbf{x},t) = g_2(\mathbf{x})\phi_\tau(t)$ with $\phi_\tau(t)$ is the cubic β-spline ($\beta^3(t)$). Moreover, these functions are deformed along the motion trajectories by applying a warping operator W [25] in order to exploit the temporal evolution of the video signal. This warping operator is an affine two-dimensional translation of the atoms. The atom index ω in case of $\mathcal{D}_{st}$ is given as: $\omega = (\mathbf{b}, \mathbf{a}, \theta, s, t_0)$, where the **a** is the spatial scale **a**, **b** is the spatial translation, θ is the rotation angle. The parameters s and t_0 are defined as the temporal scale and the temporal shift respectively, according to the translation and the scaling unitary operators $\mathcal{T}$ as follows

$$\mathcal{T}_{t_0,s}\phi_\tau(t) = \frac{1}{\sqrt{s}}\phi_\tau(\frac{t-t_0}{s}) \qquad \text{with} \qquad \phi_\tau(t) = \beta^3(t). \tag{31}$$

The function center, t_0, sweeps the entire group of pictures (GOP). The temporal scale is logarithmically discretized according to $s = s_0^i$, with s_0 is fixed and equal to 2, and

$$i \in \{0, \ldots, \lfloor \log_2(Size_{GOP}) \rfloor\}.$$

B. Spatio-temporal Dictionary Partitioning

The spatio-temporal dictionary $\mathcal{D}_{st}$ is partitioned based on its spatial component into L sub-dictionaries $\mathcal{D}_i$. The partitioning is achieved through the space-based or the region-based partitioning methodology similarly to the one described for images but with some modifications to take into account the temporal component. Let $I(\mathbf{x},t_0)$ refer to the frame at t_0, which is chosen from the image sequence $I(\mathbf{x},t)$ according to some specific criteria to denote the *reference* frame. This reference frame $I(\mathbf{x},t_0)$ of size (X_1, X_2) is divided into a set of non-overlapping rectangular blocks of size $(X_{1_\Omega}, X_{2_\Omega})$. Every block is associated to a region $\Omega_i(t_0)$. The dictionary partitioning $\mathcal{P}_{\Omega(t_0)}(\mathcal{D}_{st})$ of $\mathcal{D}_{st}$ is defined as the collection of L disjoint sub-dictionaries $\mathcal{D}_i$. Each sub-dictionary $\mathcal{D}_i$ is composed of all the warped spatio-temporal atoms $W(g_\omega(\mathbf{x}-\mathbf{x}_0, t-t_0))$ such that their spatial center $\mathbf{x}_0$ at the reference frame t_0 belongs to the unique region $\Omega_i(t_0)$. The sub-dictionary $\mathcal{D}_i$ can be interpreted as a stream of all trajectories traversing the region $\Omega_i(t_0)$.

So in more formal terms, the sub-dictionary $\mathcal{D}_i$ is defined as

$$\mathcal{D}_i = \{W(g_\omega(\mathbf{x}-\mathbf{x}_0, t-t_0)) : \mathbf{x}_0 \in \Omega_i(t_0)\}$$

with the following two conditions:

$$\Omega(t_0) = \bigcup_{i=1}^{L} \Omega_i(t_0),$$
$$\Omega_i(t_0) \bigcap \Omega_j(t_0) = \emptyset, \qquad \text{for} \qquad i \neq j.$$

An example of two spatio-temporal atoms $W(g_{\omega_i})$ and $W(g_{\omega_j})$ belonging to two sub-dictionaries $\mathcal{D}_i$ and $\mathcal{D}_j$ respectively is shown in Figure 9.

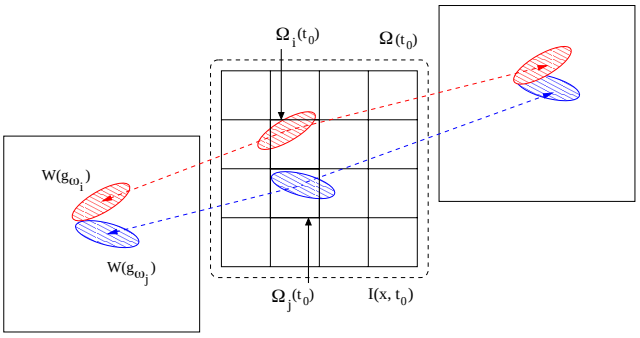


Fig. 9. An illustration of space-based partitioning in the three-dimensional case. The sequence frame $I(\mathbf{x},t_0)$ or the domain $\Omega(t_0)$ is divided into a set of separate rectangular regions $\{\Omega_i(t_0)\}$. And consequently, each one is associated with a sub-dictionary $\mathcal{D}_i$, composed of all warped atoms $W(g_\omega(\mathbf{x}-\mathbf{x}_0, t-t_0))$ centered in $\Omega_i(t_0)$ at frame $I(\mathbf{x},t_0)$. For instance, the two warped atoms $W(g_{\omega_i})$ (in red) and $W(g_{\omega_j})$ (in blue) are defined to belong to the sub-dictionaries $\mathcal{D}_i$ and $\mathcal{D}_j$ respectively.

C. The MTP Video Decomposition

The MTP algorithm, when applied to an image sequence $I(\mathbf{x},t)$, generates recursively an approximation $\tilde{I}(\mathbf{x},t)$ that is given as a linear combination of some selected warped atoms, i.e.,

$$\tilde{I}(\mathbf{x},t) = \sum_{j=1}^{J}\sum_{i=1}^{M_j} c_{j,i} W(g_{\omega_{j,i}}(\mathbf{x},t)).$$

Let us denote the residual signal at iteration j as $R^j I(\mathbf{x},t)$. Again by convention, $R^0 I(\mathbf{x},t)$ stands for the original signal $I(\mathbf{x},t)$. The algorithm performs two steps at every iteration j: the selection step and the projection step. Both steps are described in details in Algorithm 3.

Selection Step: In this step, the MTP looks for M *good* atoms where at most one atom is selected from each sub-dictionary by employing the same search strategy used in [23]. First, the reference frame t_0 is chosen dynamically from the residual signal $R^jI(\mathbf{x},t)$ as the one having the largest energy, i.e.,

$$t_0 \stackrel{def}{=} \arg\left(\max_{\tau\in GOP}\|R^jI(\mathbf{x},\tau)\|_2\right).$$

Once t_0 is selected, a dictionary partition of $\mathcal{D}_{st}$ is defined $\mathcal{P}_{\Omega(t_0)}(\mathcal{D}_{st})$. The fact that the three-dimensional atom is constituted from the temporal evolution of the spatial component implies that there exists a natural ordering in the search policy for a good atom. First, the spatial component must be located then it is evolved over time. However, propagating the best spatial atom that is present at a given frame t_0 over time does not necessarily result in the best spatio-temporal atom. Nevertheless, it is likely that the atom constructed in such a way is close to the optimal. Thus, all inner products between spatial atoms and the reference frame $R^jI(\mathbf{x},t_0)$ are computed in order to locate the L candidate spatial atoms. Every candidate $g_{\omega_i}(\mathbf{x})$ must have the largest inner product amplitude in the region $\Omega_i(t_0)$. Then, each one of the L spatial atoms is used to build the warped three-dimensional atoms $W(g_{\omega_i}(\mathbf{x})\phi_s(t-t_0))$ that are aligned on the motion trajectories according to W. After that, the atom $W(g_{\lambda_i}(\mathbf{x},t))$ is selected as the one that has the largest inner product amplitude with the residual signal $R^jI(\mathbf{x},t)$ among all possible temporal scales s in $\mathcal{D}_i$. Finally, only M_j atoms are admitted into the set Φ_Λ following the two criteria: satisfying the thresholding condition (γ) and by imposing that the coherence $\mu_1(M_j-1)$ in the collection of the selected atoms Φ_Λ does not exceed an established threshold μ_s. In other words, the following condition must be met for all admitted atoms.

$$\frac{\mu_1(M_j-1)}{M_j-1}\leq\mu_s.$$

The process for atoms admission into the set Φ_Λ is done according to the non-decreasing order of their inner product magnitudes, which is $|\langle R^jI(\mathbf{x},t),W(g_\lambda)\rangle|$, for the same reason that was already mentioned in image decomposition through the MTP algorithm.

Algorithm 3 The MTP-based Decomposition Algorithm.

INPUT:

The image sequence $I(\mathbf{x},t)$, the number of iterations J, the threshold γ and μ_s.

OUTPUT:

The set of selected atoms $\mathcal{S}$, the approximation a_J and residual R^JI.

PROCEDURE

1. Initialize the residual signal $R^0I=I$, the approximation $a_0=0$, the set of atoms $\Phi_\Lambda=\emptyset$, the set $\mathcal{S}=\emptyset$ and the iteration number $j=1$.

2: Select a *reference* frame t_0 having the largest energy in $R^jI(\mathbf{x},t)$ and set $M_j=L$.

3. For $i=1$ to L

Search for

$$g_{\omega_i}(\mathbf{x})=\arg\sup_{\omega\in\Omega_i(t_0)}|\langle R^{j-1}I(\mathbf{x},t_0),g_\omega(\mathbf{x})\rangle|.$$

Find the three-dimensional atom $W(g_{\lambda_i})$ such that:

$$W(g_{\lambda_i})=\arg\sup_{\forall s}|\langle R^{j-1}I(\mathbf{x},t),W(g_{\omega_i}(\mathbf{x})\phi_s(t-t_0))\rangle|,$$

$$\Phi_\Lambda\leftarrow\Phi_\Lambda\cup\{W(g_{\lambda_i})\}.$$

4: Discard $W(g_{\lambda_j})$ from Φ_Λ and decrement M_j if its coherence $(\mu_1(M_j-1)/(M_j-1))>\mu_s$ in Φ_Λ or

$$|\langle R^{j-1}I,W(g_{\lambda_j})\rangle|<\gamma\sup_{\lambda\in\Lambda}|\langle R^{j-1}I,W(g_\lambda)\rangle|$$

following the non-decreasing order of $|\langle R^{j-1}I,W(g_\lambda)\rangle|$.

5. Determine the orthogonal projection $P_VR^{j-1}I$ onto the span of Φ_Λ, update $\mathcal{S}$, the approximation and the residual:

$$\begin{aligned}\mathcal{S}&\leftarrow\mathcal{S}\cup\Phi_\Lambda,\\ a_j&\leftarrow a_{j-1}+P_VR^{j-1}I,\\ R^jI&\leftarrow R^{j-1}I-P_VR^{j-1}I.\end{aligned}$$

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

D. Approximation and Coding Performance

We compare here the performance of the proposed solution to several competitor schemes. The embedded quantization and coding of the atoms and their coefficients based on the subset approach, proposed for MP encoders [23], has been applied in the MTP scheme in order to generate compressed bitstreams. The motion vectors are coded losslessly by using DPCM and an adaptive arithmetic coding. The parameters of MTP have been set to $\gamma=0.5$ and $\mu_s=0.01$. The rate-distortion performance of the MTP-based video compression scheme are further evaluated against the performance of video compression standards like H.264 [26] and MPEG-4 [27], a hybrid scheme based on Matching Pursuit (MP-UCB) [28] and our previous MP-based video codec (the MP3D scheme), which is a highly scalable coding scheme and it has very interesting R-D performance at low and medium rate. In all these schemes, we set the group of picture (GOP) size to be 16, for instance in the H.264 scheme, we used one I-frame and 15 P-frames in each GOP. The reason behind setting GOP size to 16, is to have highly scalable schemes, random access to GOPs and error resilience.

The standard test sequences of Football SIF and Foreman CIF have been used in the experiments. Figure 10 shows the graphs of the rate-distortion curves of both image sequences. It is first observed that the quality of MTP decomposition is lower than that of the MP by a margin of 0.2-0.7 dB for the same number of selected atoms, as expected. The PSNR of MTP-based video codec is higher than the one of MPEG-4 over all the range of low and medium bit rates, for both sequences. However, it is less than the one of H.264 by about 1.0 dB over the entire range under study. One can check also that the MTP-based video codec performs slightly worse than the MP3D scheme. Of course, we do not expect

improvements in compression performances since the primary reason for introducing the MTP algorithm is to reduce the computational complexity of the MP algorithm. Nevertheless, this study confirms that the important reduction of complexity does not come at the expense of an important loss in coding performance. Finally, we see the that the hybrid codec MP-UCB provides similar performance as the MTP scheme at medium rate, while it is penalized by its block structure at low rate.

VII. Conclusions

Compact representation of signals over generic dictionares $\mathcal{D}$ requires the selection of the primitives that best match the signal components. Actually, this problem is well studied in approximation theory and it is known as the problem of sparse approximation in unrestricted dictionaries, whose optimal solution is *NP-hard* in general. Greedy approaches, such as the MP and the OMP algorithms have been proposed to provide sub-optimal solutions by choosing iteratively one atom at a time. However, the computational complexity of these algorithms is often cumbersome and limits their applicability. To alleviate this issue, we introduce a new greedy algorithm called the M-Term Pursuit (MTP), whose approximation performance is only slightly lower than those of MP but with large gain in computational complexity. MTP jointly selects groups of atoms in a dictionary that has been partitioned a priori in incoherent sub-dictionaries. We further illustrate the application of the MTP algorithm to image and video representation. The resulting decompositions lead to compressed bitstream whose rate-distortion characteristics come close to MP based solutions, which have been shown to compete favorably with classical encoders at low to medium rate. However, MTP offers an important reduction computation complexity. 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 such classes of signals. Another possible extension lies in the design of improved algorithms for dictionary partitioning, such as scale-based or phase-based partitioning.

References

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

[2] 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.

[3] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *27th Annual Asilomar Conference on Signals, Systems and Computers*, November 1993.

[4] A. Rahmoune, P. Vandergheynst, and P. Frossard, "The m-term pursuit for image representation and progressive compression," in *IEEE ICIP*, Genova, Italy, September 2005.

[5] P. Jost, P. Vandergheynst, and P. Frossard, "Tree-based pursuit: algorithm and properties," *IEEE Transactions on Signal Processing*, vol. 54, no. 12, pp. 4685–4697, December 2006.

[6] D. L. Donoho, Y. Tsaig, I. Drori, and J. luc Starck, "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit," Tech. Rep., 2006.

[7] M. E. Davies and T. Blumensath, "Faster & greedier: Algorithms for sparse reconstruction of large datasets," in *International Symposium on Communications, Control, and Signal Processing, ISCCSP*, 2008, pp. 774–779.

[8] D. Needell and J. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2006.

[9] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[10] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[11] D. Gabor, "Theory of communication," *J. IEE*, vol. 93, pp. 429–457, 1946.

[12] K. Gröchenig, *Foundation of time-frequency analysis*. Boston: Birkhäuser, 2001.

[13] I. Daubechies, *Ten Lectures on Wavelets*. SIAM, 1992.

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

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

[16] ——, "Topics in sparse approximation," Ph.D. dissertation, Computational and Applied Mathematics, UT-Austin, August 2004.

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

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

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

[20] 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.

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

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

[23] A. Rahmoune, P. Vandergheynst, and P. Frossard, "Flexible motion-adaptive video coding with redundant expansions," *IEEE Transactions on Circuit and Systems for Video Technology*, vol. 16, no. 2, pp. 178–190, 2006.

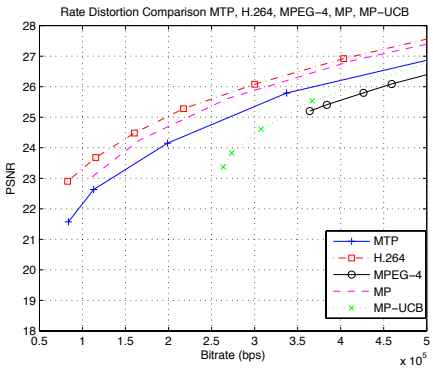
[24] R. M. Figueras i Ventura, P. Vandergheynst and P. Frossard, "Low rate and scalable image coding with redundant representations," *IEEE Transactions on Image Processing*, vol. 15, no. 3, pp. 726–739, March 2006.

[25] A. Rahmoune, "Image/video representation and scalable coding using redundant dictionaries," Ph.D. dissertation, Signal Processing Laboratory, Ecole Polytechnique Fédérale de Lausanne (EPFL), 2005.

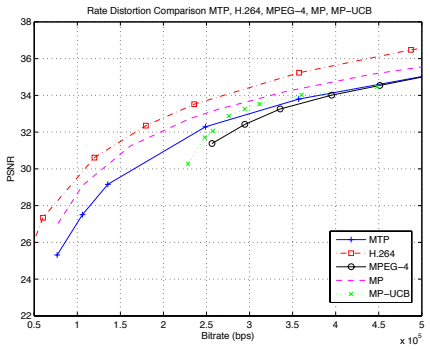
[26] H.264/AVC, "Reference software, http://bs.hhi.de/ suehring/tml/."

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

[28] R. Neff and A. Zakhor, "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.



(a) Football SIF



(b) Foreman CIF

Fig. 10. Rate distortion comparison between the MTP-based video codec ($\gamma = 0.5$, $\mu_s = 0.01$), MP3D, MP-UCB, H.264 and MPEG-4 for Football SIF and Foreman CIF sequences.