EURASIP Journal on Applied Signal Processing 2004:17, 2705–2714

Hybrid Video Coding Based on Bidimensional Matching Pursuit

Lorenzo Granai
Signal Processing Institute, Swiss Federal Institute of Technology (EPFL), 1015 Lausanne, Switzerland
Email: lorenzo.granai@epfl.ch

Emilio Maggio
Department of Electronic Engineering, Queen Mary University of London, London E1 4NS, UK
Email: emilio.maggio@elec.qmul.ac.uk

Lorenzo Peotta
Signal Processing Institute, Swiss Federal Institute of Technology (EPFL), 1015 Lausanne, Switzerland
Email: lorenzo.peotta@epfl.ch

Pierre Vandergheynst
Signal Processing Institute, Swiss Federal Institute of Technology (EPFL), 1015 Lausanne, Switzerland
Email: pierre.vandergheynst@epfl.ch

Received 23 February 2004; Revised 1 July 2004; Recommended for Publication by Mark Liao

Hybrid video coding combines together two stages: first, motion estimation and compensation predict each frame from the neighboring frames, then the prediction error is coded, reducing the correlation in the spatial domain. In this work, we focus on the latter stage, presenting a scheme that profits from some of the features introduced by the standard H.264/AVC for motion estimation and replaces the transform in the spatial domain. The prediction error is so coded using the matching pursuit algorithm which decomposes the signal over an appositely designed bidimensional, anisotropic, redundant dictionary. Comparisons are made among the proposed technique, H.264, and a DCT-based coding scheme. Moreover, we introduce fast techniques for atom selection, which exploit the spatial localization of the atoms. An adaptive coding scheme aimed at optimizing the resource allocation is also presented, together with a rate-distortion study for the matching pursuit algorithm. Results show that the proposed scheme outperforms the standard DCT, especially at very low bit rates.

Keywords and phrases: image processing, greedy algorithms, matching pursuit, redundant dictionaries, video coding, H.264/AVC.

1. INTRODUCTION

The most successful class of video compression algorithms is based on hybrid methods consisting in the combination of prediction loops in the temporal dimension (motion estimation/motion compensation) with a suitable uncorrelation technique in the spatial domain (transform coder).

The state of the art for hybrid video coding is specified by the recent standard H.264, also named advanced video coding (AVC) (ITU-T Rec. H.264, or ISO MPEG-4, part 10). In this work, we aim at exploiting the advantages of coding the displaced frame difference (DFD), output of the motion compensation (MC) algorithm, using a redundant dictionary. This kind of dictionaries leaves more freedom to the basis functions design and therefore they can be created with the goal of catching the structures of DFDs. In order to remain as close as possible to the state of the art, we adopt a motion estimation algorithm that is compatible with H.264 (see Section 2). The output of this block is then coded using a pursuit algorithm and an appositely designed bidimensional, anisotropic dictionary. Thanks to this technique, we achieve a sparse representation of the signal and therefore a more compact energy concentration.

The problem of recovering the sparsest representation over a given redundant dictionary corresponds to the minimization of the l_0 norm of the representation. In general, this is a nonpolynomial (NP) problem, but recent results show that, under certain conditions on signal and dictionary, the sparsest solution can be approximated using greedy techniques such as orthogonal matching pursuit (OMP) [1] or matching pursuit (MP) [2, 3].

This kind of methods experiences an increasing success especially for one-dimensional signal representation (e.g., see [4]) and natural image representation [5, 6]. MP has been already used for video coding too: for example, in [7, 8], the authors present an MP-based codec which offers very good performances. The main differences with respect to this method are the use of a dictionary of bidimensional, nonseparable, anisotropic functions, the atom selection performed through the entire frame, and the coding technique.

The main points of our work are

(a) the design of a redundant dictionary suitable for coding DFD,
(b) the use of fast techniques for atom selection, which work in the Fourier domain and exploit the spatial localization of the atoms,
(c) the adaptive coding scheme aimed at optimizing the resource allocation for transmitting the atom parameters,
(d) the rate-distortion (RD) study for the MP algorithm which allows an optimal selection of the number of atoms to be placed in every frame.

In addition, the obtained results are compared with a technique that codes the DFD (found using the same MC) using a classical DCT scheme and with the standard H.264.

This paper is structured as follows. Section 2 presents the motion estimation/compensation block, inspired by the H.264/AVC standard. The coding algorithm adopted for DFDs is explained in Section 3, with details about new faster methods for atom selection. Section 4 illustrates the in-loop quantization and entropy coding, while the RD optimization is explained in Section 5. Results and comparisons can be found in Section 6, while Section 7 concludes and presents possible future developments.

2. MOTION ESTIMATION

High compression efficiency in video coding is achieved by adopting hybrid systems which combine two stages. In the first stage, motion estimation and MC predict each frame from the neighboring frames. At the second one, the prediction error is coded. Current video compression standards use block-based orthogonal transforms to code the residual error. These two stages are then followed by appropriate entropy coding.

Relative to prior coding methods, the standard H.264/AVC has an enhanced motion estimation that allows higher compression ratios [9]. In particular, we can attribute this improvement to the new variable block-size MC with small block sizes, the quarter-sample-accurate MC, and the use of multiple reference frames. Moreover, the 4×4 integer transform turns out to be well adapted to this kind of MC [10].

In our coding scheme, we adopt some of the new features introduced by this standard and obtain an MC scheme that is compatible with H.264. In particular, we used the following features:

(i) variable block-size MC, with a minimum size of 4×4,
(ii) tree-based MC,
(iii) MC with quarter-pel accuracy,
(iv) use of improved "skipped" motion inference [9].

Our encoder allows I- and P-pictures only. Moreover, due to the frame-based structure of our MP codec, intrablocks are not permitted. I-pictures are fully compliant with the H.264/AVC standard, using the integer transform illustrated in [10]. Currently, only three of the nine prediction directions are used and only the 4×4 predicted block mode is implemented (not the 16×16 one) [9].

3. CODING DISPLACED FRAME DIFFERENCES

The residual error of the motion compensated prediction still contains spatial redundancy: to reduce the amount of resources needed for transmission, this error is typically coded via block-based DCT. In H.264/AVC, this transform is replaced by an integer orthogonal approximation of the DCT, able to work with 4×4 blocks and so compatible with the finest MC segmentation. The advantage of this transform is that it can be computed exactly in integer arithmetic, thus avoiding inverse transform mismatch problems; moreover, it reduces the computational complexity thanks to the fact that it can be calculated without multiplications, in 16-bit arithmetic [10].

However, linear invariant block-based transforms are far from optimal for representing (and then compressing) bidimensional signals such as natural images or motion compensated images [11]. In [7, 8, 12], the authors have shown that improved coding efficiency can be achieved by replacing the DCT with an overcomplete nonorthogonal transform. This kind of approach, together with a suitable dictionary design, can represent a valid alternative to DCT or wavelet-based schemes, especially (but not necessarily only) at low bit rates, where most of the signal energy can be captured by only a few elements of the dictionary.

In the proposed scheme, the output of the motion estimation is a predicted image that is subtracted from the current frame. The DFD, difference between these two images, is then coded with an MP algorithm, as explained in the following. Note that this algorithm is not block-based: both the coding and the atom selection procedures work on the full frame, without any spatial subdivision.

3.1. Greedy algorithms

Structured signals can be effectively represented by a superposition of few elements selected from a specifically designed redundant dictionary of basis functions. We then say that such signals have a sparse representation over the dictionary $\mathcal{D}$. Once we have designed an "appropriate" dictionary to decompose our structured signal, if we are able to find the sparsest representation or sparsest m-term approximation, it follows that we are representing the signal in the most efficient way. In general, this leads to an efficient compression.

We take a signal f which has a sparse decomposition b over the dictionary $\mathcal{D}$ such that

$$f = \mathcal{D}b = \sum_{g_i \in \Lambda} g_i b_i, \tag{1}$$

where Λ is a subset of $\mathcal{D}$, with $|\Lambda| = m$. The problem of finding the sparsest solution of (1) corresponds to minimizing the l_0 norm of the representation, $\|b\|_0$. In the general case, it is an NP hard problem. However, recent results show that under certain conditions on the dictionary and the signal, the problem can be solved with linear complexity. The first results, given by Donoho and Xuo in [13] and Elad and Bruckstein in [14], discuss the uniqueness of the sparsest solution and the independence from the sparseness measure. In practice, the solution can be found by minimizing $\|b\|_1$, the l_1 norm of (1), which leads to the basis pursuit principle [15].

The latest results in [2, 3, 16] prove that the greedy algorithms MP and OMP can also recover sparse solutions and moreover they can achieve a sparse approximation of the signal with an exponential decay of the energy of the error. It is important to notice that the condition of incoherence introduced by Donoho and Elad et al. is a bit relaxed with "quasi-incoherent dictionary," a concept developed by Tropp, that permits to prove the good behavior of basis pursuit and MP with more redundant dictionaries. Taking into account the good approximation property of MP and the flexibility that it allows concerning the dictionary design, we think that this greedy decomposition algorithm could be a good candidate in order to code structured signals, especially at low bit rates. It is worth mentioning that, compared with the OMP decomposition or with the linear programming used to solve the basis pursuit problem, MP allows solutions that make it faster.

3.2. Matching pursuit

In this subsection, we recall the basics of the iterative process used for the selection of the waveforms that represent the signal structures. A more detailed explanation of the MP algorithm can be found in [17].

Let $\mathcal{D} = \{g_\gamma\}_{\gamma \in \Gamma}$ be a dictionary of unitary norm vectors g_γ called atoms and let Γ represent the set of possible indexes. At the Nth iteration, a function f is decomposed as follows:

$$f = \sum_{n=0}^{N-1} \langle g_{\gamma_n}, R^n f \rangle g_{\gamma_n} + R^N f, \tag{2}$$

where $R^0 f = f$ and $R^n f$ is the residual after the nth step. To minimize the residual, at each iteration, we must choose g_{γ_n} such that the absolute value of the projection $|\langle g_{\gamma_n}, R^n f \rangle|$ is maximal. It can be proved [17] that $R^n f$ converges exponentially to zero when n tends to infinity. Since at each iteration, the residual and the selected atom are orthogonal, it follows that

$$\|f\|^2 = \sum_{n=0}^{N-1} |\langle g_{\gamma_n}, R^n f \rangle|^2 + \|R^N f\|^2. \tag{3}$$

Equation (3) expresses the energy conservation of MP. The convergence depends on both the dictionary and the search strategy. In [18], it has been shown that there are two real numbers $\alpha, \beta \in]0, 1]$ such that for all $n \geq 0$, the following relation is valid:

$$\|R^{n+1} f\| \leq (1 - \alpha^2 \beta^2)^{1/2} \cdot \|R^n f\|, \tag{4}$$

where α is an optimality factor related to the strategy adopted to select the best atom in the dictionary, while β depends on the dictionary, representing its ability to capture the features of the input function f [19].

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

$$k \cdot N \cdot d \cdot n \log_2 n, \tag{5}$$

where d depends on the size of the dictionary (it is actually the size of the dictionary without considering translations), N is the number of chosen atoms, and the constant k depends on the strategy adopted for atom selection. In particular, we can obtain $k \ll 1$. See also Section 3.4 where we propose two solutions to speed up the atom selection and pick up more than one atom per iteration. Given a highly redundant dictionary, MP proves to be more computationally demanding than both the 8×8 DCT and the 4×4 integer transform used in H.264, whose complexity is $O(n \log_2 n)$.

3.3. Dictionary design

Dictionary design is a crucial item for MP, since it strongly affects its convergence and visual performances. The dictionary used in our experiments is particularly suited for exploiting the signal structures of DFDs, mainly thanks to the use of peculiar generating functions and anisotropy (see also [20]).

The proposed dictionary is thus composed of a set of real bidimensional functions, named atoms, built by applying the following three types of transformations to the generating function $g(\vec{x}) : \mathbb{R}^2 \to \mathbb{R}$ with $\vec{x} = (x_1, x_2)$.

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

$$\mathcal{T}_{\vec{b}} g(\vec{x}) = g(\vec{x} - \vec{b}). \tag{6}$$

(b) Rotation $\mathcal{R}_\theta$, to locally orient the atom:

$$\mathcal{R}_\theta g(\vec{x}) = g(r_\theta(\vec{x})), \tag{7}$$

where r_θ is a rotation matrix:

$$r_\theta(\vec{x}) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{8}$$

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

$$\mathcal{S}_{\vec{a}} g(\vec{x}) = \mathcal{S}_{a_1, a_2} g(x_1, x_2) = g\left(\frac{x_1}{a_1}, \frac{x_2}{a_2}\right). \tag{9}$$

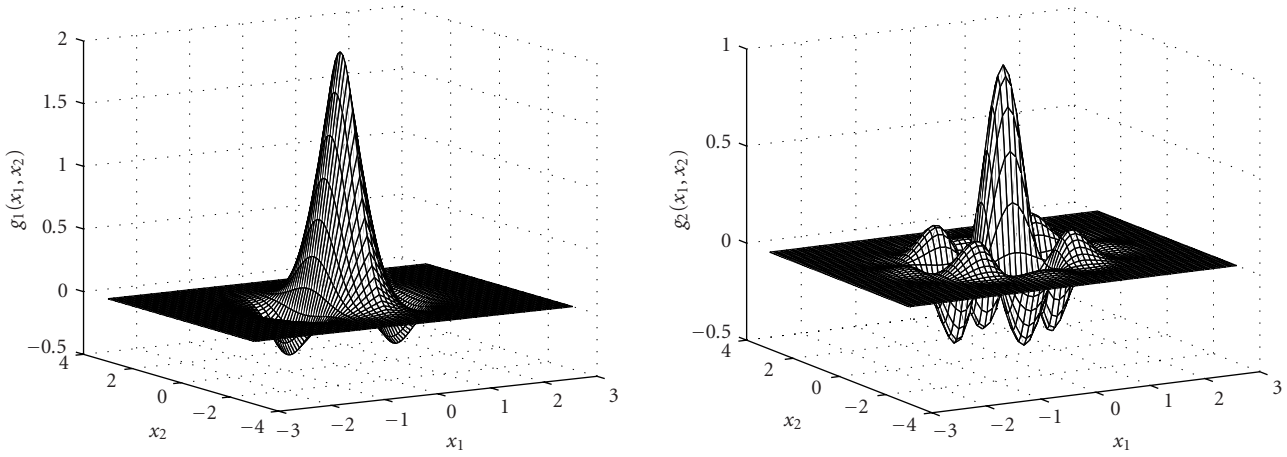

Figure 1: Generating functions g_1 and g_2.

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

$$\text{atom}_{(\vec{b},\theta,\vec{a})}(\vec{x}) = \mathcal{T}_{\vec{b}}\mathcal{R}_{\theta}\mathcal{S}_{\vec{a}}g(\vec{x}). \tag{10}$$

Finally, the obtained waveforms are normalized as follows:

$$\text{atom}^{\text{norm}}_{(\vec{b},\theta,\vec{a})}(\vec{x}) = \frac{\text{atom}_{(\vec{b},\theta,\vec{a})}(\vec{x})}{||\text{atom}_{(\vec{b},\theta,\vec{a})}(\vec{x})||_2}. \tag{11}$$

The dictionary used by the MP algorithm is obtained by suitably discretizing all parameters:

$$\mathcal{D} = \left\{\text{atom}^{\text{norm}}_{(\vec{b},\theta,\vec{a})}(\vec{x})\right\}_{\vec{b},\theta,\vec{a}} \tag{12}$$

In [5], it has been shown that bended atoms can improve the performances of an MP encoder when the target is a natural still picture. We tested this option for video signals, finding that only an extremely small gain in terms of error and visual quality is obtained, but with the drawback of a big increase in the dictionary size. Thus, we choose not to include this transformation in our set.

The "mother functions" which generate the whole dictionary with the previous transformations have been selected in order to best match the characteristics of the input signal, that is, the DFD coming out from the MC block. In particular, three functions have been chosen.

(a) A second derivative of a B-spline on the x_1 axes, times a bivariate exponential; see (13) and Figure 1. It is a peaky function that fits the usual behavior of DFDs; this function is nothing else than a small variation of the piecewise function introduced in [20] for coding motion-compensated prediction errors:

$$g_1(x_1, x_2) = g_{bs}(x_1)e^{-(x_1^2+x_2^2)}, \tag{13}$$

where g_{bs} is as follows:

$$g_{bs}(x) = \begin{cases} -2 + 3|x| & \text{if } 0 \le |x| < 1, \\ 2 - |x| & \text{if } 1 \le |x| < 2, \\ 0 & \text{if } |x| \ge 2. \end{cases} \tag{14}$$

(b) A Gabor function with oscillations in both the x_1 and the x_2 directions and with a frequency independent of the scaling factors (see Figure 1). Note that this function has an additional parameter for the frequency but has only two possible rotations that correspond to the vertical and horizontal positions:

$$g_2(x_1, x_2) = \cos(\omega_x x)\cos(\omega_y y)e^{-(x_1^2+x_2^2)}. \tag{15}$$

In our implementation, we set $\omega_x = \omega_y$.

(c) A simple rectangular function expressed by (16), able to code errors due to the block-based nature of the MC:

$$g_3(x_1, x_2) = \begin{cases} 1 & \text{if } |x_1| < 1 \wedge |x_2| < 1, \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

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

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

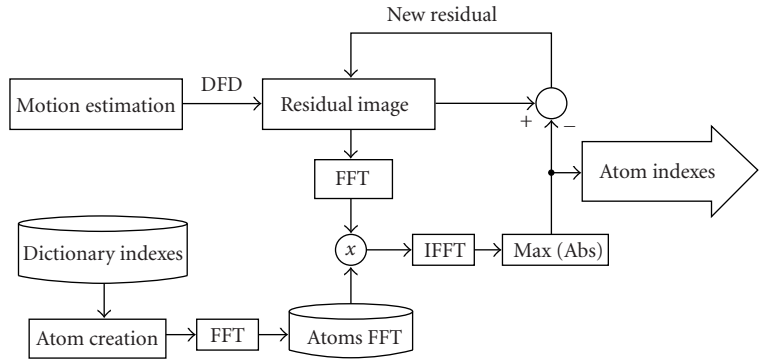

Figure 2: Scheme for the atom selection in the Fourier domain.

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

$$\mathcal{D} = \left\{\text{atom}_{(g,\vec{b},\theta,\omega,\vec{a})}(\vec{x})\right\}_{g,\vec{b},\theta,\omega,\vec{a}}. \tag{17}$$

Here the index g specifies which function has been chosen to create the atom, ω is the frequency used only for the Gabor functions, while the other values are the same as in (12). Finally, the number of waveforms in our dictionary (the parameter d in (5)) is approximately 1000: each of them can additionally be translated in any location of the image (see (6)). This set of atoms proves to be highly redundant.

3.4. Atom selection

MP decomposes a DFD into its most important features: this greedy algorithm, as previously described, selects at each iteration an atom from the dictionary such that the projection coefficient $|\langle g_{\gamma_n}, R^n f\rangle|$ is maximum. To find such g_{γ_n}, we use a full-search algorithm that computes the inner products between the residual and all the functions of the dictionary. Since the dictionary is composed of all the translations of the transformed generating functions (TGFs), see (10), it is clear that all the inner products between the TGF translated all over the residual and the residual itself correspond to the convolutions of the TGF with the residual. In order to speed up the search, convolutions are computed like products in the frequency domain, as depicted in Figure 2; the Fourier transform of the entire dictionary is computed only once at the beginning of the video sequence and stored. Direct and inverse Fourier transforms are computed in a fast way using the FFTW package (http://www.fftw.org/) (version 3.0.1, see [21]).

Even with this method, the atom selection is still too slow for our purposes. Here we propose two solutions to speed up the algorithm. The first method (multiple atom algorithm), already introduced in [5], consists of a slightly modified version of MP: at each iteration, more than one atom is selected and used to decompose the residual. This can be done since in an image, there are structures that are definitely separated in the spatial domain, and this is even more evident in a DFD where the features to code are usually small. Like in (2), we can write

$$f = \sum_{k=0}^{K-1} \left(\sum_{n=n_k}^{n_{k+1}-1} \langle g_{\gamma_n}, R^n f\rangle g_{\gamma_n} \right) + R^N f, \tag{18}$$

with $n_0 = 0$ and $n_k = N$. At the kth iteration, all the atoms of the dictionary are sorted according to the absolute value of the projections. Starting from the one with the highest projection, all the n_k atoms that are quasiorthogonal are selected. Selecting on average $\overline{n}_k$ atoms at once, it turns out that MP only needs $N/\overline{n}_k$ iterations. For example, decomposing a QCIF sequence, we observed a speed-up factor of around 10. The drawback of this method is that there is no more guaranty that at each iteration, the best atom will be selected as in the case of the full-search MP. However, the resulting loss in the image quality is almost negligible.

A second possible strategy to speed up the searching algorithm can be found considering that from one iteration to another, usually only a small area of the residual image changes. At the first iteration, all the convolutions between the image and each atom are computed; the main idea of this method is to store these values and at the next iteration update them only in the region where the best atom has been placed. The gain lies in performing the convolution and the inverse Fourier transform on a smaller area. The gain increases as the selected atoms get smaller (have a smaller surface). This solution is possible only because the atoms we are using have a limited spacial support, as already observed in Section 3.3. This method has no quality loss and, according to our simulations, gives a gain in computational time of around 20% compared with the full search in the Fourier domain [22]. On the other hand, the required memory increases around 30%.

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

4. QUANTIZATION AND ENTROPY CODING

As said in Section 3.3, parameters that specify an atom in the dictionary are the generating function type, two scale factors, the rotation angle, and, only for Gabor atoms, the frequency. Moreover, we have to add to this list the atom position (two natural numbers whose range is determined by the frame size) and its projection coefficient. The indexes that characterize the atom shape are entropy coded using an adaptive arithmetic coding algorithm. Since the rotation depends on the x_2-scale, the arithmetic algorithm uses the conditioned probability $p(\text{rotation}|x_2\text{-scale})$ to code the rotation parameter.

In order to code the positions and projection coefficients of the atoms, two different approaches can be taken into account. The first one consists in ordering the atoms according to their decreasing projection absolute values, then the projections are quantized in a differential way (DPCM) followed by arithmetic coding; the x_1 and x_2 coordinates are simply stored without any particular coding scheme. We will refer to this scheme as "projection DPCM" coding. The second approach performs a different sorting of the atoms in such a way to take advantage of coding the atoms positions [12], coding the coordinates in a differential way, followed by arithmetic coding. We will refer to this scheme as "position" coding. Another interesting approach for coding the atoms is presented in [23], where bit-plane quantization of atom projections and quadtree prediction of atom positions are combined.

For both "projection DPCM" and "position" coding, quantization is performed in-loop: this provokes the reinjection of quantization error in the coding loop and permits encoding of this error. For a detailed study about in-loop quantization for MP, we recommend [24]. Yet, we have to emphasize that our approach is independent and does not follow the modelization that is proposed in the cited paper.

4.1. "Position" versus "projection" coding

At very low bit rates, when just few atoms per frame are coded, the projection DPCM method gives the best results. When the number of atoms per frame increases, the position encoding improves and finally outperforms the projection DPCM; later, the gap between these two coding styles increases together with the number of atoms selected (see Figure 3). This phenomenon is easily explicable, since the position DPCM performances are related to the atoms density in the frame.

For example, simulations showed that for QCIF sequences, usually the switching point is around 50 atoms/frame, after this threshold, position encoding starts to outperform projection DPCM. With 200 atoms/frame, the average gain is around 10% of the rate [22]. Figure 4 shows the percentage of bits allocated to code the atom's parameters, positions, and projections in both cases.

4.2. An adaptive solution

The situation illustrated by Figure 3 suggests that we can optimize the coding procedure by running both the previously illustrated entropy encoders and choosing the best one. In practice, after the position coding has been selected for few consecutive iterations, we can stop checking and start to use this method only. In this way, we always adopt the best coding solution, and from a rate point of view, the only price to be payed is absolutely negligible: one bit per frame to specify the coding style. The possibility to switch from one encoding method to an other is integrated in the RD optimization, explained in next section.

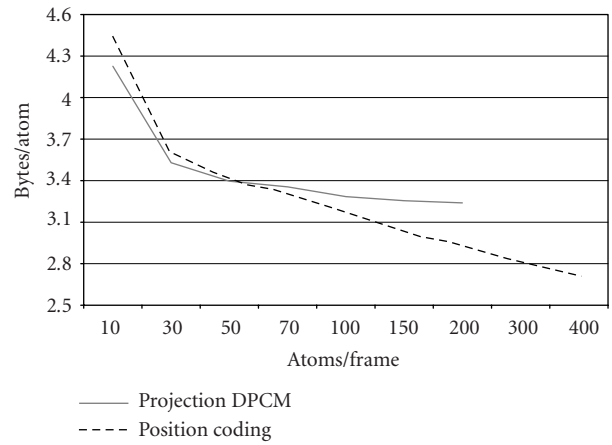

Figure 3: Bytes per atom necessary to code 19 frames of Container QCIF using different encoding styles.

5. RATE-DISTORTION OPTIMIZATION

In a video sequence, some consecutive frames are very similar one to each other: in this case, the DFD contains very few information and, in our MP implementation, it can be coded with a small number of atoms. On the other hand, there are situations in which the amount of information to code strongly increases, requiring more atoms. Hence, given a certain target bit rate, or a fixed quality, we have to face the problem of choosing the number of atoms per frame. A classical approach to this kind of issues is based on the minimization of a Lagrangian RD functional [25]:

$$\min\{J\}, \quad J = D + \lambda R, \ \lambda \geq 0. \tag{19}$$

In (19), D is the distortion (MSE) and R is the rate (bytes/second); λ is constant for the whole sequence. For a convex problem, the necessary and sufficient condition to find the absolute minimum of J is

$$\frac{\partial D}{\partial n} = -\lambda \frac{\partial R}{\partial n}. \tag{20}$$

The first term in (20) is the variation of MSE through iterations, a negative number whose value is linked to the energy of the residual that an atom is able to catch. The second term represents the weighted differential rate. We can state that $\partial R/\partial n$ is always positive and on average decreases with n. Hence $-\lambda(\partial R/\partial n)$ is negative and increases. In order to minimize J, we need a last consideration: the two terms of

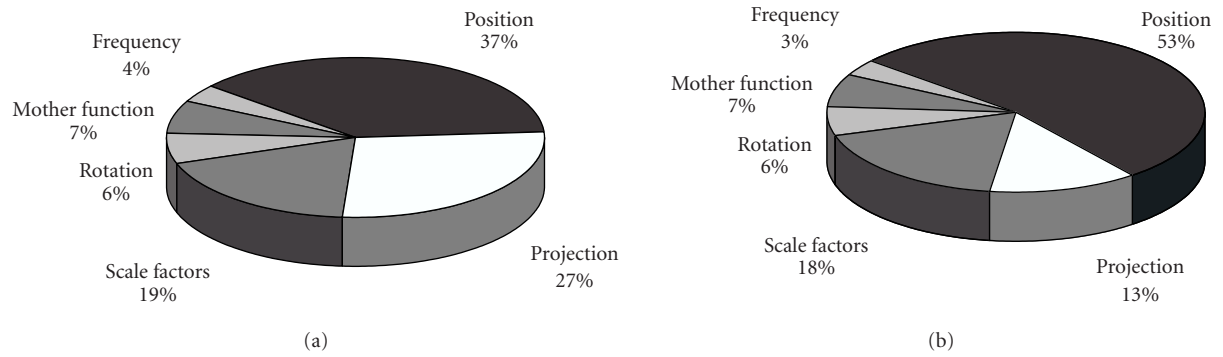

Figure 4: Example of typical bit allocations for (a) position and (b) projection DPCM encoding styles.

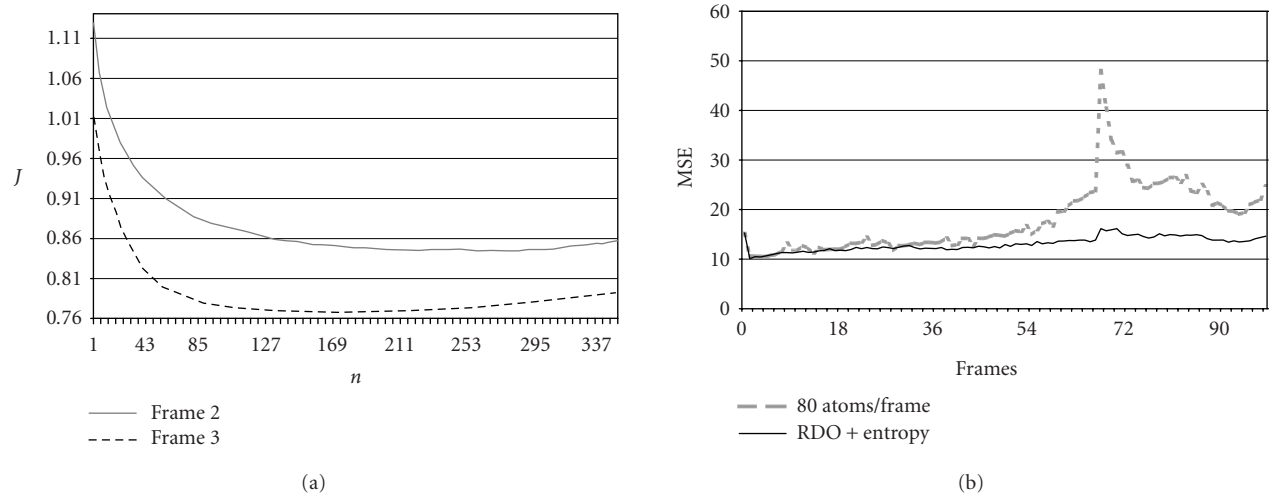

Figure 5: (a) RD optimization: $J(n)$ for two frames of Stefan. (b) MSE for the first 100 frames of News coded with and without RD optimization.

(20) are both negative and they increase on average with decreasing the first derivative, but their limit when $n \to \infty$ is different (the first limit comes from [17, Lemma 2]):

$$\lim_{n\to\infty} \frac{\partial D}{\partial n} = 0, \qquad \lim_{n\to\infty} -\lambda \frac{\partial R}{\partial n} = C. \tag{21}$$

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

$$\lim_{n\to 0} \frac{\partial D}{\partial n} < \lim_{n\to 0} -\lambda \frac{\partial R}{\partial n}, \tag{22}$$

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

$$\lim_{n\to 0} \frac{\partial D}{\partial n} \geq \lim_{n\to 0} -\lambda \frac{\partial R}{\partial n}, \tag{23}$$

and we have to stop the expansion when the condition in (20) is respected. From (21), thanks to the continuity of the first derivative of R and D, and assuming that both $\partial R/\partial n$ and $-\lambda(\partial R/\partial n)$ with their first derivatives are monotonically decreasing (and not only in average), it comes that there exists only one point $\tilde{n}$ which solves (20) and this point is the absolute minimum we are looking for. In theory, since the dictionary is finite, the constant C in (21) can assume the value 0, a depending solution adopted for coding the atoms. Anyway this situation has no practical interest since we never use a number of atoms which can be comparable with the size of the dictionary.

From an implementation point of view, we have the problem that the differential MSE has a monotone trend but it does not always increase with n. The same observation holds for the differential rate. These small deviations from the ideal behavior imply the possible existence of local minima. However, this problem can be easily solved, since $J(n)$ always shows a precise trend, as can be seen in Figure 5a. The only precaution we take is not to stop the coding process exactly when J starts to increase, but to go on for few iterations in order to be sure that we are not in a local minimum.

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

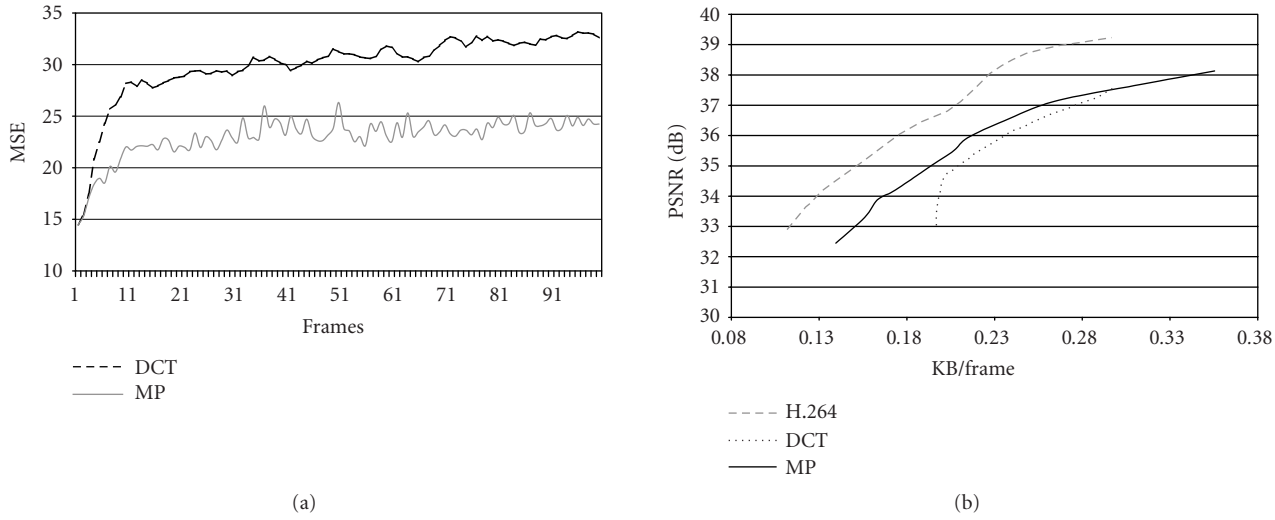

Figure 6: (a) MSE obtained by coding the first 100 frames of Container using MP (0.190 KB/frame) and the 8×8 DCT (0.194 KB/frame); no I-frames. (b) RD curves obtained by coding the first 100 frames of Traffic using 8×8 DCT and MP with the same motion estimation and H.264; I-frames enabled.

It is important to point out that this RD approach can be used even when the atom selection is performed by turning to the multiple atom algorithm (see Section 3.4). In this case, however, some changes are required due to the fact that atoms are not necessarily selected in decreasing order of projection absolute value. Hence at the first step, we subtract all the selected atoms from the residual but we code only the best one, and we put all the others in a list sorted by decreasing projections. In the following steps, we code the best of the current step plus all the atoms in the list whose projection is higher than the projection of the best atom of the current step.

In order to compute the rate, two different situations have to be taken into account since we do not know a priori if a position or projection DPCM coding style will be adopted (see Section 4). Also the choice between these methods is then left to the RD algorithm.

Figure 5b shows the MSE behavior of the test sequence "News." It is easy to observe the improvement achieved by the RD optimization with respect to the case in which a fixed number of atoms per frame is coded.

6. RESULTS AND COMPARISONS

The first comparisons are aimed at testing the quality of the MP codec with respect to a standard 8×8 DCT. So we adopt the same motion estimation described in Section 2 and we then code the DFDs using either MP or a classical DCT block-based scheme. The MP atom selection is performed using the fast multiple atom algorithm, explained in Section 3.4. In this case, for all the tested sequences, the MP outperforms DCT. For example, Figure 6a shows the MSE behavior for the sequence "Container" in QCIF format: even if the DCT has a slightly higher rate, it is outperformed by MP in terms of both visual quality and mean square error. In Figure 6b, one can see the RD curve obtained by coding a video-surveillance traffic sequence (QCIF format), allowing the encoders to put I-frames when necessary. Comparisons show the superiority of MP versus DCT, especially at very low bit rates. Moreover, thanks to several algorithm optimizations [22], a real-time decoding is possible for sequences up to CIF format.

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

(i) Hadamard transform: enabled;
(ii) search range: 16;
(iii) number of reference frames: 1;
(iv) block sizes (for motion estimation): all enabled;
(v) B-frames: disabled;
(vi) CABAC: disabled.

Results clearly show that H.264 obtains better performances than our encoder. For example, coding the sequence "Traffic" in QCIF format, we can observe a gap of more than 1.5 dB (see Figure 6b). This gap can be explained assuming that the H.264 encoder is fully optimized for the block-based integer transform, while we work in a frame-based way. In fact, we notice that, especially at low bit rates, the losses due to a coding syntax not suited for the overall coder heavily affect the performances of MP. We also have to consider that, even with some disabled option, the motion estimation of H.264 is still more accurate than the one we used in our MP implementation (see also Section 2). In fact, we did not disable all the features missing in our MC algorithm and this results in a not completely fair comparison between the two approaches.

7. CONCLUSIONS

In this paper, we present a new video coding scheme based on H.264 motion estimation and bidimensional MP. The use of a redundant dictionary allows to design basis functions that catch the main structures of a DFD so that a sparse representation of the signal is obtained. Atom selection is performed on the whole frame, with a fast algorithm. Atom parameters are quantized in-loop and entropy coded, using an adaptive criterion to choose which encoding style best fits the atoms stream. A rate distortion optimization is performed in order to select the number of atoms per frame. Simulations at very low bit rates show that, given the same motion estimation algorithm, MP outperforms 8×8 DCT. If this proves the superiority of the proposed scheme versus more standard transform techniques, on the other hand, it is not sufficient to equal the performances of the standard H.264. This is mainly due to a lack of optimization between the MC part and the DFD coding.

The approach we present here, being based on MP, implies a computational cost that is definitely higher than the standards. Nevertheless, it involves many advantages, like the possibility of easily including scalability, the improved visual quality, and the flexibility in the dictionary design. The latter point can in particular be exploited by optimizing the dictionary or adapting it to the changes of the residual image [26]: for example, when there are no more edges, we could deactivate the B-spline and rectangular functions, inserting new, smaller atoms. Moreover, good suboptimal strategies (here we propose two of them) can considerably reduce the complexity of the MP algorithm.

More work on the quantization of the projection value would be necessary. In fact, for the position entropy coding mode, we have used a simple uniform quantizer, while finding more appropriate ways to reduce the range of the quantized values could improve the compression ratio. In addition, an RD system which takes into account also the quantization step of the atoms could improve the coding efficiency.

ACKNOWLEDGMENTS

The authors would like to thank Guillaume Baud for his active collaboration and Alessandro Mecocci for his help in this project. Many thanks to Markus Flierl for his useful pieces of advice and to Fulvio Moschetti and Pascal Frossard for interesting discussions and comments. This work has been partly supported by the SNF Grants 2100-066912.01/1 and NCCR IM.2 and by Visiowave S.A. (CTI Grant 6044.1 KTS).

REFERENCES

[1] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proc. 27th IEEE Asilomar Conference on Signals, Systems & Computers*, vol. 1, pp. 40–44, Pacific Grove, Calif, USA, November 1993.

[2] J. Tropp, "Greed is good: Algorithmic results for sparse approximation," Tech. Rep., Texas Institute for Computational Engineering and Sciences, Austin, Tex, USA, 2003.

[3] R. Gribonval and M. Nielsen, "Approximation with highly redundant dictionaries," in *Proc. SPIE Wavelets: Applications in Signal and Image Processing X*, vol. 5207 of *Proceedings of SPIE*, pp. 216–227, San Diego, Calif, USA, August 2003.

[4] R. Gribonval and E. Bacry, "Harmonic decomposition of audio signals with matching pursuit," *IEEE Trans. Signal Processing*, vol. 51, no. 1, pp. 101–111, 2003.

[5] L. Peotta, L. Granai, and P. Vandergheynst, "Very low bit rate image coding using redundant dictionaries," in *Proc. SPIE Wavelets: Applications in Signal and Image Processing X*, vol. 5207 of *Proceedings of SPIE*, pp. 228–239, San Diego, Calif, USA, August 2003.

[6] P. Frossard, P. Vandergheynst, and R. Figueras i Ventura, "High-flexibility scalable image coding," in *Proc. SPIE Visual Communications and Image Processing (VCIP '03)*, vol. 5150 of *Proceedings of SPIE*, pp. 127–134, Lugano, Switzerland, July 2003.

[7] R. Neff and A. Zakhor, "Matching pursuit video coding. I. Dictionary approximation," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 1, pp. 13–26, 2002.

[8] R. Neff and A. Zakhor, "Matching-pursuit video coding. II. Operational models for rate and distortion," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 1, pp. 27–39, 2002.

[9] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[10] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 598–603, 2003.

[11] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, New York, NY, USA, 1998.

[12] O. K. Al-Shaykh, E. Miloslavsky, T. Nomura, R. Neff, and A. Zakhor, "Video compression using matching pursuits," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 9, no. 1, pp. 123–143, 1999.

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

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

[15] S. S. Chen, *Basis Pursuit*, Ph.D. thesis, Department of Statistics, Stanford University, Stanford, Calif, USA, 1995.

[16] R. Gribonval and P. Vandergheynst, "On the exponential convergence of matching pursuit in quasi-incoherent dictionaries," submitted to *IEEE Transactions on Information Theory*.

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

[18] P. Vandergheynst and P. Frossard, "Efficient image representation by anisotropic refinement in matching pursuit," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP '01)*, vol. 3, pp. 1757–1760, Salt Lake City, Utah, USA, May 2001.

[19] P. Frossard and P. Vandergheynst, "Redundancy in non-orthogonal transforms," in *Proc. IEEE International Symposium on Information Theory (ISIT '01)*, p. 196, Washington, DC, USA, June 2001.

[20] F. Moschetti, L. Granai, P. Vandergheynst, and P. Frossard, "New dictionary and fast atom searching method for matching pursuit representation of displaced frame difference," in *Proc. IEEE International Conference on Image Processing (ICIP '02)*, vol. 3, pp. 685–688, Rochester, NY, USA, June 2002.

[21] M. Frigo and S. G. Johnson, "FFTW: an adaptive software architecture for the FFT," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP '98)*, vol. 3, pp. 1381–1384, Seattle, Wash, USA, May 1998.

[22] E. Maggio, "Un nuovo schema di codifica video con matching pursuit basato su una compensazione del moto H.264 compatibile," M.S. thesis, Universitá degli studi di Siena, Siena, Italy, 2003.

[23] J. Lin, W. Hwang, and S. Pei, "SNR scalability based on bitplane coding of matching pursuit atoms at low bit rates," submitted to *IEEE Trans. Circuits and Systems for Video Technology*.

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

[25] A. Ortego and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, 1998.

[26] Y.-T. Chou, W.-L. Hwang, and C.-L. Huang, "Gain-shape optimized dictionary for matching pursuit video coding," *Elsevier Signal Processing*, vol. 83, no. 9, pp. 1937–1943, 2003.

Lorenzo Granai was born in Siena, Italy, on February 18, 1975. He received the M.S. degree in telecommunication engineering from the University of Siena, Italy, in 2001. Since 2002, he has been a Ph.D. student and Research Assistant at the Signal Processing Institute, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. In 2002, he visited the Multimedia Signal Processing Laboratory, NTT DoCoMo, Inc., Japan, where he worked as a Guest Researcher in the area of video coding. His current research interests are in signal processing, image representation and coding, and approximation theory.

Emilio Maggio received the M.S. degree in telecommunication engineering from Information Engineering Department, University of Siena, Siena, Italy, in 2003. He is currently pursuing the Ph.D. degree at the Electronic Engineering Department, Queen Mary University, London. He is with the Digital Signal Processing & Multimedia Research Group. His research interests include video coding, Bayesian inference, and signal processing and tracking, with particular emphasis on particle filters and mean shift approaches.

Lorenzo Peotta was born in Vicenza, Italy, on October 21, 1973. He received the M.S. degree in telecommunication engineering from the University of Padova, Italy, in 1999. From 2000 to 2001, he attended the Doctoral School in Communication Systems at the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. Since 2002, he has been a Ph.D. student and Research Assistant at the Signal Processing Institute, EPFL. His current research interests are in image and video coding, nonlinear signal approximation/representation, approximation theory, and information theory.

Pierre Vandergheynst received the M.S. degree in physics and the Ph.D. degree in mathematical physics from the Université Catholique de Louvain, Louvain, Belgium, in 1995 and 1998, respectively. From 1998 to 2001, he was a Postdoctoral Researcher with the Signal Processing Laboratory, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. He is now an Assistant Professor of visual information representation theory at EPFL, where his research focuses on computer vision, image and video analysis, and mathematical techniques for applications in visual information representation. Dr. Vandergheynst is Co-Editor-in-Chief of Signal Processing.