Hybrid video coding using bi-dimensional matching pursuit

Lorenzo Granai, Emilio Maggio, Lorenzo Peotta and Pierre Vandergheynst

Abstract

In this report we propose a new coding scheme based on the Matching Pursuit algorithm and exploiting some of the new features introduced by H.264 for motion estimation. Main points of this work are the design of a redundant dictionary suitable for coding displaced frame differences, the use of fast techniques for atom selection, which work in the Fourier domain and exploit the spatial localization of the atoms, the adaptive coding scheme aimed at optimizing the resouce allocation for transmitting the atom parameters and the Rate-Distortion optimization.

Index Terms

Video coding, H.264/AVC, Greedy algorithms, Matching Pursuit, Redundant dictionaries, Rate-Distortion optimization

I. Introduction

MOTION video data consist essentially of a time-ordered sequence of pictures. 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 (ITU-T Rec. H.264, or ISO MPEG-4, part 10), also named Advanced Video Coding (AVC). The advantage of H.264 with respect to other compression techniques is achieved thanks to several improvements that this standard introduces. We can individuate the main items in the new variable block-size motion compensation with small block sizes, the quarter-sample-accurate motion compensation, the use of multiple reference frames, the 4x4 integer transform, the in-loop deblocking filter and the context adaptive binary arithmetic coder.

In this work we intent to profit of some of the features that H.264 introduced for motion estimation, but replacing the transform in the spatial domain, such that the Displaced Frame Difference (DFD) will be coded using a pursuit algorithm and an appositely designed bi-dimensional, 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 Non-Polynomial (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) or Matching Pursuit (MP) [1], [2].

This kind of methods knows an increasing success expecially for one-dimensional signal representation (for example see [3]) and natural image representation [4], [5]. MP has been already used for video coding too: for example in [6], [7] authors present an encoder based on Matching Pursuit which offers very good performances. Main differences with respect to this method are the use of a dictionary of bi-dimensional, non-separable, anisotropic functions, the atom selection performed through the entire frame and the coding technique.

The main points of our work are:

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

This report is structured as follows: Section II presents the motion estimation/compensation block, inspired by the H.264/AVC standard. The coding algorithm adopted for displaced frame differences is explained in Section III, with details about new faster methods for atom selection. Section IV illustrates the in-loop quantization and entropy coding, while the Rate-Distortion optimization is explained in Section V. Results and comparisons can be found in Section VI, while Section VII presents conclusions and possible future developments.

http://lts2www.epfl.ch
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)

II. 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 compensation 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 [8]. In our coding scheme, we adopt some of the new features introduced by this standard and obtain a motion compensation scheme that is compatible to H.264. In particular we used the following features:

- variable block-size motion compensation (MC), with a minimum size of 4x4,
- tree-based MC,
- MC with quarter-pel accuracy ,
- use of improved “skipped” motion inference [8].

Our encoder permits I and P-pictures only. Moreover, due to the frame-based structure of our Matching Pursuit encoder, intra-block are not permitted. I-pictures are fully compliant with the H.264/AVC standard, using the integer transform illustrated in [9]. Currently, only three of the nine prediction directions are used and only the 4x4 predicted block mode is implemented (not the 16x16 one) [8].

III. 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 4x4 blocks and so compatible with the finest motion compensation segmentation. The advantage of this transform is that it can be computed exactly in integer arithmetic, so 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 (see [10], [9]).

However, these linear invariant block-based transforms are far from optimal for representing (and then compressing) bi-dimensional signals such as natural images or motion compensated images [11]. In [12], [13] authors have shown that improved coding efficiency can be achieved by replacing the DCT with an overcomplete non-orthogonal 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 difference between these two images is then coded with a Matching Pursuit 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.

A. Greedy Algorithms

Structured signals can be very 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-terms approximation, it follows that we are representing the signal in the most efficient way. In general this leads to an efficient compression.

Let us take a signal f which has a sparse decomposition b over the dictionary $\mathcal{D}$,

$$b \quad \text{s.t.} \quad f = \mathcal{D}b = \sum_{g_i \in \Lambda} g_i b_i, \tag{1}$$

where Λ is a subset of $\mathcal{D}$, with $|\Lambda| = m$. The problem of finding the sparsest solution of equation (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. First results, given by Donoho et al. in [14] and Elad et al. in [15], 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 [16].

Latest results in [1], [2] 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 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 rate. It is worth mentioning that, compared with the OMP decomposition algorithm or with the linear programming used to solve the Basis Pursuit problem, Matching Pursuit allows solutions that make it faster.

B. 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 Matching Pursuit 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 N^{th} 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 n^{th} 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:

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

Eqn. (3) expresses the energy conservation of MP. The convergence of MP 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].

In subsection III-D one can find other methods that we propose in order to speed up the selection of the atomwith the highest projection and to select more than one atom per iteration.

C. Dictionary Design

Dictionary design is a crucial item for the MP algorithm since 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 anysotropy (see also [20]).

The proposed dictionary is so composed of a set of real bi-dimensional 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:

$$\mathcal{T}_{\vec{b}}\ g(\vec{x}) = g(\vec{x} - \vec{b}). \tag{5}$$

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

$$\mathcal{R}_\theta\ g(\vec{x}) = g(r_\theta(\vec{x})), \tag{6}$$

where r_θ is a rotation matrix

$$r_\theta(\vec{x}) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{7}$$

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

$$\mathcal{S}_{\vec{a}}\ g(\vec{x}) = \mathcal{S}_{a_1,a_2}\ g\left(x_1, x_2\right) = g\left(\frac{x_1}{a_1}, \frac{x_2}{a_2}\right). \tag{8}$$

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

$$\text{atom}_{(\vec{b},\theta,\vec{a})}(\vec{x}) = \mathcal{T}_{\vec{b}}\ \mathcal{R}_\theta\ \mathcal{S}_{\vec{a}}\ g(\vec{x}). \tag{9}$$

Finally the obtained waveforms are normalized as follows:

$$\text{atom}^{\text{norm}}_{(\vec{b},\theta,\vec{a})}(\vec{x}) = \frac{\text{atom}_{(\vec{b},\theta,\vec{a})}(\vec{x})}{\|\text{atom}_{(\vec{b},\theta,\vec{a})}(\vec{x})\|_2}. \tag{10}$$

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

$$\mathcal{D} = \{\text{atom}^{\text{norm}}_{(\vec{b},\theta,\vec{a})}(\vec{x})\}_{\vec{b},\theta,\vec{a}} \; . \tag{11}$$

In [4] 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 of 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, i.e. the DFD coming out form the motion compensation block. In particular three functions have been chosen:

- A second derivative of a B-Spline on the x_1 axes, times a bivariate exponential, see eq. (12). 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:

  $$g_1(x_1, x_2) = g_{bs}(x_1)e^{-\left(x_1^2+x_2^2\right)}, \tag{12}$$

  where g_{bs} is

  $$g_{bs}(x) = \begin{cases} -2 + 3\,|x| & \text{if} \quad 0 \leq |x| < 1 \\ 2 - |x| & \text{if} \quad 1 \leq |x| < 2 \\ 0 & \text{if} \quad |x| \geq 2 \end{cases} \; . \tag{13}$$

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

  $$g_2(x_1, x_2) = cos(\omega_x x)cos(\omega_y y)e^{-(x_1^2+x_2^2)}. \tag{14}$$

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

- A simple rectangular function expressed by Eqn. (15), able to code errors due to the block-based nature of the motion compensation:

  $$g_3(x_1, x_2) = \begin{cases} 1 \quad \text{if} \quad |x_1| < 1 \wedge |x_2| < 1 \\ \\ 0 \qquad\quad \text{otherwise} \end{cases} \; . \tag{15}$$

  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 vertical and horizontal.

The whole the dictionary is composed by 2D atoms, computed in a non-separable way. Moreover spatial supports of all the waveforms are limited since, when 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.

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

$$\mathcal{D} = \{\text{atom}_{(g,\vec{b},\theta,\omega,\vec{a})}(\vec{x})\}_{g,\vec{b},\theta,\omega,\vec{a}} \; . \tag{16}$$

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 equation (11). Finally the number of waveforms that compose our dictionary is approximately 1000: each of them can additionally be translated in any location of the image (see Eqn. (5)). This set of atoms results to be highly redundant.

D. Atom Selection

Matching Pursuit 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 by all the translations of the transformed generating functions (TGF), see Eqn. (9), it is clear that all the inner products between the TGFs translated all over the residual and the residual itself correspond to the convolutions of the TGF with the residual. To speed up the search we compute the convolutions like products in the frequency domain, as depicted in Fig. III-D; 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 (version 3.0.1, [21], [22]).

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 [4], consists into a slightly modified

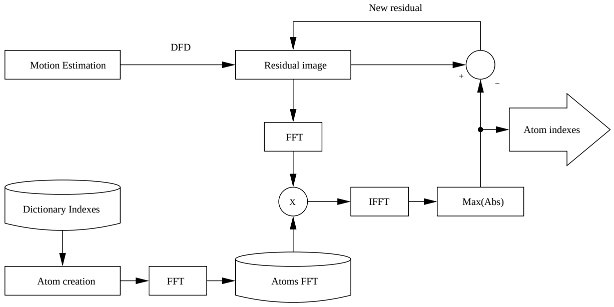

Fig. 1. Scheme for the atom selection in the Fourier domain

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 Eqn. (2), we can write:

$$f = \sum_{k=0}^{K-1} \left(\sum_{n=n_k}^{n_{k+1}-1} \langle g_{\gamma_n}, R^n f \rangle g_{\gamma_n} \right) + R^N f, \tag{17}$$

with $n_0 = 0$ and $n_K = N$. At the k^{th} iteration, according to the absolute values of the projection coefficients, all the atoms of the dictionary are sorted. Starting from the one with highest projection, all the n_k atoms that are quasi-orthogonal 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 image quality is almost negligible.

A second possible strategy to speed up the searching algorithm can be found considering that from one iteration to an other 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 lays on performing the convolution and the inverse Fourier transform on a smaller area. The gain increases as selected atoms get smaller (have a smaller surface). This solution is possible only because the atoms that we are using have a limited spacial support, as already observed in subsection III-C. 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 [23].

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.

IV. Quantization and Entropy Coding

As said in Section III-C, 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 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 $p(rotation|x_2\text{-}scale)$ to code the rotation parameter.

In order to code the positions and projection coefficients of the atoms, two different and non-compatible 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 in coding the atoms' positions [13], coding the coordinates in a differential way followed by arithmetic coding. We will refer to this scheme as "position" coding.

In both cases the quantization is performed in-the-loop: this provokes the re-injection of quantization error in the coding loop and permits encoding of this error.

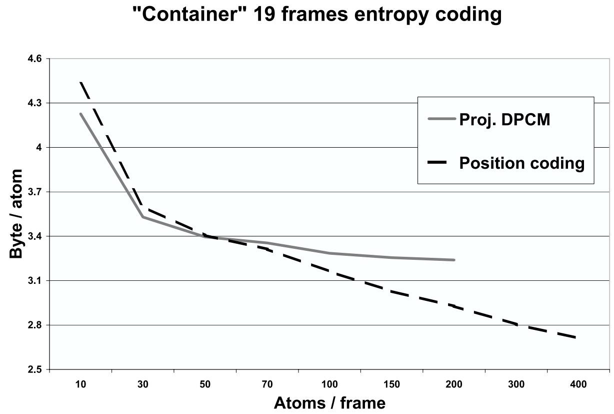

Fig. 2. Bytes per atom necessary to code 19 frames of “Container” QCIF using different encoding styles

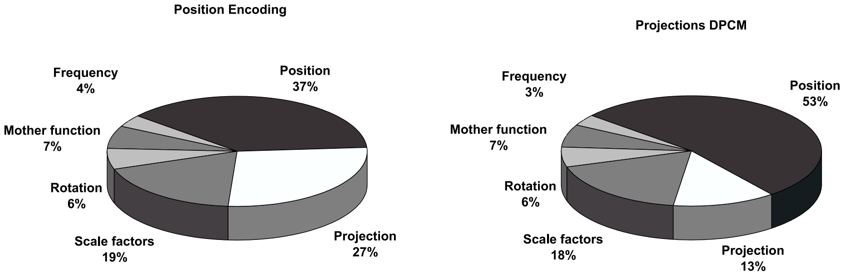

Fig. 3. Example of typical bit allocations for “position” and “projection DPCM” encoding styles

A. “Position” vs. “Projection” Coding

At very low bit rate, 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 Fig. IV-A). 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 [23]. Fig. IV-A shows the percentage of bits allocated to code the atoms parameters, positions and projections in both cases.

B. An adaptive solution

The situation illustrated by Fig. IV-A 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 Rate-Distortion (RD) optimization, explained in next section.

V. 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

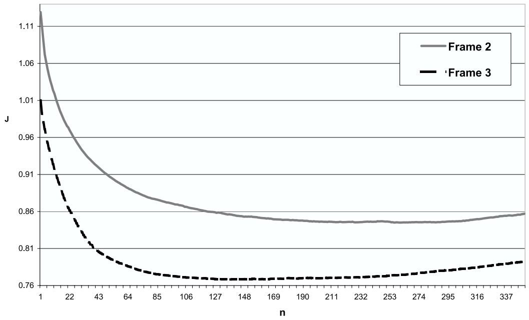

Fig. 4. Rate-Distortion optimization: J(n) for two frames of ”Stefan”.

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 rate-distortion functional [24]:

$$\min\{J\}, \quad \text{where} \quad J = D + \lambda R \quad \text{and} \quad \lambda \geq 0. \tag{18}$$

In Eqn. (18), D is the distortion (MSE) and R is the rate (byte/second). What does this equation mean for MP? Passing from the discrete domain of the atom selection to a continuous one, interpolating the values of R and D in such a way that their first derivative is continuous too, it is possible to demonstrate that, if J is twice differentiable and $\frac{\partial^2 J}{\partial n^2} > 0$ a necessary and sufficient condition to find the absolute minimum of J is:

$$\frac{\partial J}{\partial n} = \frac{\partial(D + \lambda R)}{\partial n} = 0, \tag{19}$$

where n is the index of the atom expansion, now a continuous variable. We can rewrite the previous equation as

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

The first term in eq.(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 code. The second term represents the weighted differential rate. We can state that $\frac{\partial R}{\partial n}$ is always positive and in average decreases with n. Hence, if $\lambda > 0$, $-\lambda \frac{\partial R}{\partial n}$ is negative and increases. In order to minimize J we need a last consideration: the two terms of eq.(20) are both negative and they increase in average with decreasing first derivative, but their limit when $n \to \infty$ is different (the first limit comes from lemma 2 in [17]):

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

We can easily 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 $\frac{\partial R}{\partial n}$ and $-\lambda \frac{\partial R}{\partial n}$ with their first derivatives are monotonically decreasing (and not only in average), it comes that it exists only one point $\tilde{n}$ which solves Eqn.(20) and it is the absolute minimum we are looking for.

From an implementation point of view, coming back in the discrete domain, 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 R. These small deviations from the ideal behavior imply the possible existence of local minima. This problem can be easily solved, since the behavior of $J(n)$ always shows a precise trend, as can be seen in Fig. V. The only precaution we have to take against local minima is that when J starts to increase we have to go on for few iterations in order to be sure that it is not a “false alarm”.

Moreover it is possible to use this RD approach even when the atom selection is performed turning to the multiple atom algorithm (see subsection III-D). In this case, however, some changes are required, due to the fact that atoms

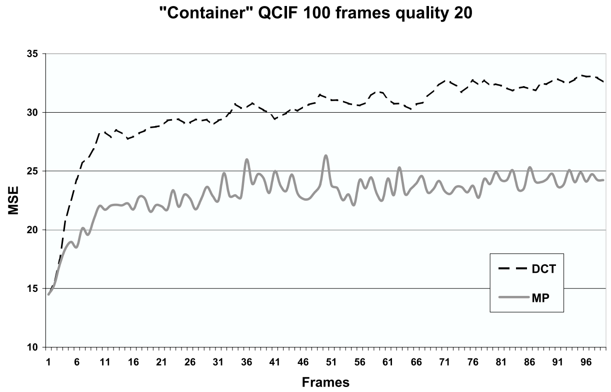

Fig. 5. MSE obtained coding the first 100 frames of “Container” using MP (0.190 KB/frame) and the 8x8 DCT (0.194 KB/frame); no key frames

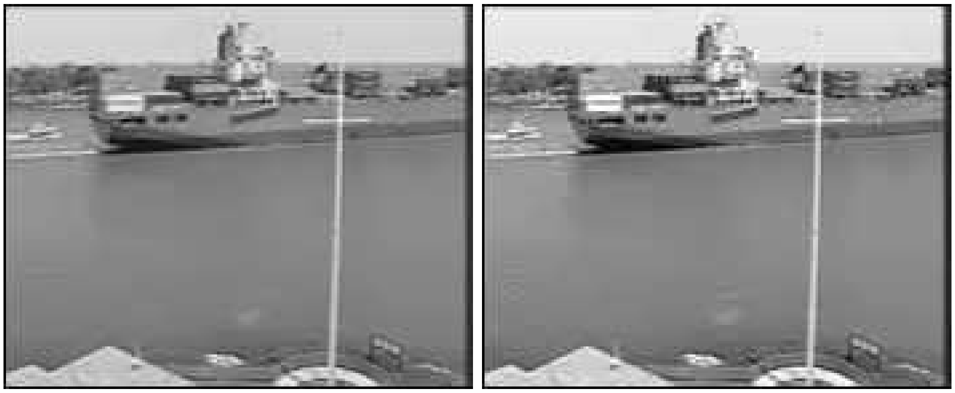

Fig. 6. 16th frame of “Container” coded with MP (left) and 8x8 DCT (right)

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 IV). Also the choice between these methods is then left to the RD algorithm. Important improvements are achieved by the RD optimization with respect to the case in which a fixed number of atom per frame is coded [23].

VI. Numerical Experiments

The first comparisons are aimed at testing the quality of the MP encoder with respect to a standard 8x8 DCT. So we adopt the same motion estimation described in Section II and we code then 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 III-D. In this case, for all the tested sequences the MP outperforms DCT. For example Fig. VI 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 (see Fig. VI too).

Fig. VI shows the R-D curve obtained by coding the first 100 frames of a video surveillance sequence, allowing the encoder to put intra frames when necessary. Comparisons show the evident superiority of MP, especially at very low bit-rates.

Moreover, thanks to several algorithm optimizations [23] a real time decoding is possible for sequences up to a CIF format.

In order to compare the MP video coder with the H.264 reference code (JM7.3, see [25]) we disabled some of the options that are not yet implemented in our motion estimation. Following settings have been used:

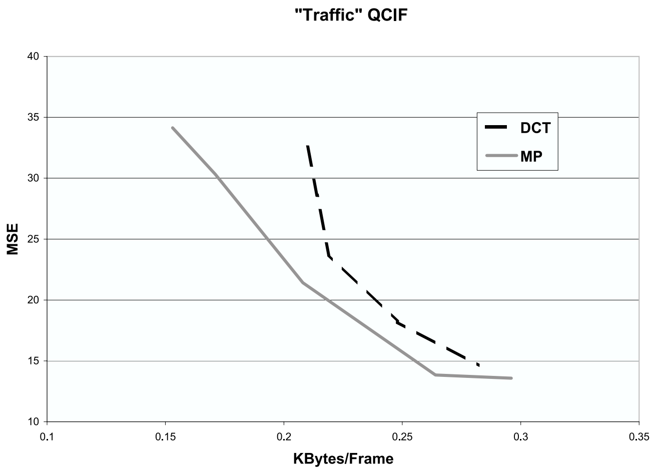

Fig. 7. Rate vs. distortion curves obtained coding the first 100 frames of “Traffic” using MP and 8x8 DCT, allowing key frame choice

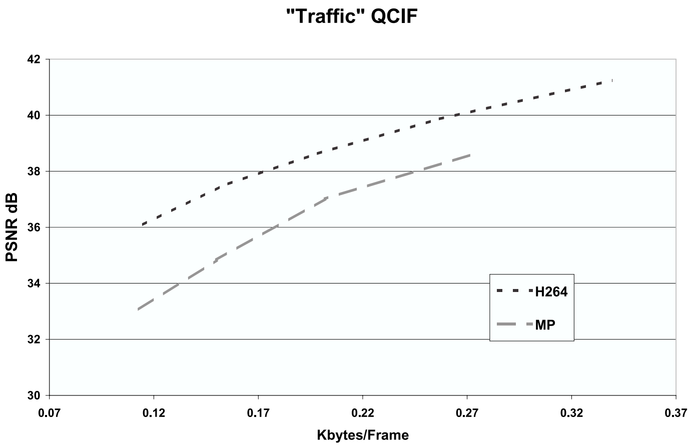

Fig. 8. RD curves obtained coding the first 100 frames of “Traffic” using MP and H.264

- Hadamar transform: enabled,
- Search range: 16,
- Number of reference frames: one,
- Block Sizes (for motion estimation): all enabled,
- B frames: disabled,
- “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 Fig. VI). This gap can be explained thinking 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 affect heavily the performances of MP. We also have to consider that, even with some disabled option, the motion estimation of the reference encoder is still more accurate than the one we used in our MP implementation (see also section II). In fact we did not disable all the features missing in our MC algorithm and this results in an advantage for the AVC performances.

VII. Conclusions

In this report we present a new video coding scheme based on H.264 motion estimation and bi-dimensional Matching Pursuit. The dictionary is designed for motion compensated images and the atom selection is performed on the full frame, with a fast full search algorithm. Atom parameters are in-loop quantized 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 show an advantage with respect to sequences coded with the same motion estimation and then DCT.

It would be interesting to find a way of adapting the dictionary to the changes of the residual image: for example when there are no more edges we could deactivate B-Spline and rectangular functions, inserting new, smaller atoms in the dictionary. More work is then necessary on the quantization of the projection values. In fact for the position entropy coding mode we have used a simple uniform quantizer, while finding clever ways to reduce the range of the quantized values could improve the compression ratio. Moreover a RD system which takes into account also the quantization step could improve the coding efficiency. A predictive Rate Distortion model for both entropy coding and quantization step has to be studied. With this model, multiple executions of the entropy coding algorithm can be avoided.

Acknowledgment

The authors would like to thank Guillaume Baud for his active collaboration and Prof. Alessandro Mecocci for his help to this project. Many thanks to Markus Flierl for his interesting and useful advices.

References

[1] J. Tropp, “Greed is good : Algorithmic results for sparse approximation,” Texas Institute for Computational Engineering and Sciences, Tech. Rep., 2003.

[2] R. Gribonval and M. Nielsen, “Approximation with highly redundant dictionaries,” in *Proc. of 48th SPIE annual meeting*, San Diego, USA, August 2003.

[3] R. Gribonval and E. Bacry, “Harmonic decomposition of audio signals with matching pursuit,” *IEEE Trans. Signal Process.*, vol. 51, no. 1, pp. 101–111, jan 2003.

[4] L. Peotta, L. Granai, and P. Vandergheynst, “Very low bit rate image coding using redundant dictionaries,” in *Proc. of 48th SPIE annual meeting*, San Diego, USA, August 2003.

[5] P. Frossard, P. Vandergheynst, and R. Figueras i Ventura, “High flexibility scalable image coding,” in *Proc. SPIE Conference on Visual Communications and Image Processing (VCIP’03)*, July 2003.

[6] R. Neff and A. Zakhor, “Matching-pursuit video coding, part I: Dictionary approximation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 13–26, Jan 2002.

[7] ——, “Matching-pursuit video coding, part II: Operational models for rate and distortion,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 27–39, Jan 2002.

[8] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 560–576, July 2003.

[9] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, “Low-complexity transform and quantization in H.264/AVC,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 598–603, July 2003.

[10] ——, “Low-complexity transform and quantization with 16-bit arithmetic for H.26L,” in *Proc. IEEE International Conference on Image Processing (ICIP’02)*, vol. 2, 2002, pp. 489–492.

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

[12] R. Neff and A. Zakhor, “Very low bit-rate video coding based on matching pursuit,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 158–171, Feb. 1997.

[13] O. K.Al-Shaykh *et al.*, “Video compression using matching pursuit,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 123–143, 1999.

[14] D. L. Donoho and X. Huo, “Uncertainty principles and ideal atom decomposition,” *IEEE Trans. Inform. Theory*, vol. 47, no. 7, pp. 2845–2862, Nov 2001.

[15] M. Elad and A. M. Bruckstein, “A generalized uncertainty principles and sparse representation in pairs of bases,” *IEEE Trans. Inform. Theory*, vol. 48, no. 9, pp. 2558–2567, Sep 2002.

[16] S. S. Chen, “Basis pursuit,” Ph.D. dissertation, Stanford University, 1995.

[17] S.Mallat and Z.Zhang, “Matching pursuit with time-frequency dictionary,” *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3397–3415, Dec 1993.

[18] P. Vandergheynst and P. Frossard, “Efficient image representation by anisotropic refinement in matching pursuit,” in *Proc. IEEE International Conference on Acoustic, Speach and Signal Processing(ICASSP’01)*, vol. 3, Salt Lake City, USA, May 2001, pp. 1757–1760.

[19] P. Frossard and P. Vandergheynst, “Redundancy in non-orthogonal transforms,” in *Proc. IEEE International Symposium on Information Theory(ISIT’01)*, 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, 2002, pp. 685–688.

[21] FFTW Home Page, http://www.fftw.org/.

[22] M. Frigo and S. Johnson, “FFTW: an adaptive software architecture for the FFT,” in *Proc. IEEE International Conference on Acoustic, Speach and Signal Processing(ICASSP’98)*, vol. 3, 1998, pp. 1381–1384.

[23] E. Maggio, “Un nuovo schema di codifica video con matching pursuit basato su una compensazione del moto H.264 compatibile,” Master’s thesis, Universita’ degli studi di Siena, Oct 2003.

[24] A. Ortega and K. Ramchandran, “Rate-distortion methods for image and video compression,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, pp. 51–73, Nov 1998.

[25] H.264/AVC Reference Software, http://bs.hhi.de/˜suehring/tml/.