Flexible Motion-Adaptive Video Coding with Redundant Expansions

Adel Rahmoune, Pierre Vandergheynst and Pascal Frossard
{adel.rahmoune,pierre.vandergheynst,pascal.frossard}@epfl.ch
Swiss Federal Institute of Technology EPFL
Signal Processing Institute ITS
CH- 1015 Lausanne, Switzerland

ITS Technical Report
ITS-TR.2004.17

Abstract

This paper presents a highly flexible video coding scheme (MP3D), based on the use of a redundant 3-D spatio-temporal dictionary of functions. Directionality and anisotropic scaling are key ingredients to the spatial components, that form a rich collection of 2-D visual primitives. The temporal component is tuned to capture most of the energy in the temporal signal evolution, along motion trajectories in the video sequences. The MP3D video coding scheme first computes motion trajectories, that are lossless entropy coded and sent as side information to the decoder. It then applies a spatio-temporal decomposition using an adaptive approximation algorithm based on Matching Pursuit (MP). Quantized coefficients and basis function parameters are entropy-coded in a embedded stream that is constructed to respect multiple rate constraints. The geometric properties of the 2-D primitive dictionary allows for flexible spatial resolution adaptation, so that the MP3D stream allows for decoding at multiple rate and spatio-temporal resolutions. The MP3D scheme is shown to provide comparable rate-distortion performances at low and medium bit rates against state-of-the-art schemes, like H.264 and MPEG-4, or the scalable MC-EZBC. It also provides an increased flexibility in stream manipulation to adapt to non-octave based spatial resolutions, or to any rate constraints. However, the use of a redundant dictionary is penalizing at high coding rates, which makes the MP3D algorithm interesting for low rate applications, or as a flexible base layer for higher rate video systems.

Index Terms

video coding, Matching Pursuit, redundant expansion, spatio-temporal atoms, scalability.

I. Introduction

High scalability video coding is becoming a stringent requirement for video streaming and delivery over the Internet and heterogeneous networks. These networks are characterized by a fluctuating channel capacity and a wide range of clients with different computational and display capabilities. Hence, the goal of highly scalable video coding is to generate a single embedded bit-stream that is able to be decoded efficiently at different bit rates and at various resolutions. This framework imposes strong constraints on the encoder: it has to operate without prior knowledge about the specific bit rate and the format at which the compressed video will be decoded. For this reason, video coding algorithms based on the predictive feedback approach, which combine motion compensation and a DCT transform, fail to achieve high scalability. An alternative method is to use a feed-forward or an open-loop approach, where a spatio-temporal transform is followed by embedded quantization and coding.

This work has been supported by the Swiss National Science Foundation.

Most scalable 3-D based approaches employ a separable 2-D wavelet transform (DWT) for the spatial information, and DWT with either a transversal or lifting implementation along motion trajectories (see Section II for more details and references). Recently however, it was pointed out that the separable 2-D wavelet transform is not ideally suited for representing images as it fails to capture regular geometric features (e.g. edges) [1].

In this paper, we elaborate a new 3-D based method that aims at overcoming these limitations. Stepping on previous work on low bit-rate image and video coding using redundant dictionaries [2]–[4], we introduce a spatio-temporal representation that consists in applying a redundant decomposition along motion trajectories. On the algorithmic point of view, using redundant dictionaries still represents quite a challenge. In Section III, we quickly review recent results on that topic. These recent advances in non-linear approximation motivate the use of the pure Greedy Algorithm (also known as Matching Pursuit [5]) as a decomposition strategy.

An overview of the MP3D coding scheme is given in Section III-C. The motion-adaptive 3-D transform is presented in details in Section IV. The algorithm comprises two steps which are detailed: the motion trajectory prediction and MP decomposition. The generation of an embedded scalable bit-stream is then presented in Section V. A rate-constrained quantization scheme is applied to the transform coefficients, that are then compressed by adaptive arithmetic coding, along with function indexes. Motion parameters are entropy-coded and sent as side information to the decoder. Experiments carried out on standard test sequences illustrate its coding efficiency in Section VI. At low and medium bit rates, (i.e. less than 500 kbps), the obtained results are very comparable to the state-of-the-art non-scalable coders H.264 and MPEG-4, and to the scalable MC-EZBC scheme in terms of rate distortion performances and visual quality. However at higher bit-rates, our redundant dictionary is less efficient in representing video data compared to those schemes. Section VII highlights special scalability features of the MP3D coding scheme: resolution and SNR scalability properties. Conclusions and perspectives are highlighted in Section VIII.

II. Related work

Most successful scalable video coding schemes are based on three-dimensional (3-D) separable discrete wavelet transform (DWT). Karlsson and Vetterli initiated the use of DWT for subband video compression [6]. Then, in the method proposed by Ohm [7], a translational model is assumed for motion, where the video frames are divided into blocks which undergoe a rigid motion. A separable 3-D wavelet transform is then applied on the displaced blocks. However, the effects of contraction and expansion in the motion field are observed by the appearance of disconnected pixels between the blocks. These disconnected pixels are handled differently, in order to make the transform invertible, which affects the overall coding performance. In the scheme proposed by Taubman and Zakhor [8], a warping operator is used in order to align the frames in the direction of motion prior to applying the 3-D transform. However, only global camera pan is treated in their warping operator in order to make it invertible. Kim and Pearlman [9] uses a 3-D separable transform, by extending the successful 2-D wavelet-based SPIHT algorithm [10] into the temporal dimension. However, without motion compensation, it produces some annoying ghosting artifacts at low bit rates, because of the temporal filtering. This motivates the exploitation of motion within the spatio-temporal transform. Subsequently, Choi and Woods [11] propose a motion-compensated 3-D scheme where a hierarchical block matching motion estimation algorithm with pruning is used with half-pel precision for displacements. In the same manner as in [7], this scheme applies a special processing for disconnected pixels, that enhances coding efficiency. It is worth noting that both warping-based and block displacement-based methods often use Haar wavelets. When longer filters are selected, they do no bring any significant improvement in terms of coding efficiency. Recently, 3-D wavelet based methods have been redesigned by employing the lifting scheme in the temporal dimension in order to have perfect reconstruction regardless of the motion model. The LIMAT scheme [12] employs a lifting implementation of the DWT, in which each lifting step is compensated for the estimated scene motion. Mesh-based motion estimation algorithms are shown to perform better than hierarchical block-based ones. Another lifting-based scheme has been also presented in [13] that satisfies the invertibility property.

In summary, most of the existing scalable video compression schemes employ a separable 2-D wavelet transform for the spatial information and a temporal DWT along the motion trajectories with either a transversal or a lifting implementation. Our approach differs from the afore-mentioned schemes in that it uses an overcomplete dictionary for adaptive signal decomposition in the spatial and temporal dimensions, instead of an orthogonal or biorthogonal one. This method allows to achieve competitive performance at low and medium bit-rates while ensuring scalability and flexibility of the compressed bit-streams. These properties however usually come at a price, which is a higher computational complexity.

III. Matching Pursuit Expansion of Image Sequences

A. Benefits of Non-linear Sparse Approximations

Most acclaimed technical solutions to both image and video compression, namely the JPEG2000 and MPEGx/H.26x families of standards, rely heavily on transform coding. Moving to the transform domain is classically performed in order to obtain decorrelated sets of coefficients on which scalar quantization and entropy coding is performed. The choice of the transform is thus driven by its de-correlating performances as well as good properties under quantization and ease of entropy coding. Most techniques use two well controlled orthonormal basis (ONB): DCT and wavelets. Performing the transform by means of an ONB allows the use of well studied data compression results, and in both cases fast algorithms help keeping a low complexity algorithm. Unfortunately, restricting a representation to an ONB fixes a very rigid structure on the components of the signals that are represented, and sometimes dramatically damages the coherence and quality of important visual primitives. This results in annoying artifacts at low bit rates on textures and edges. To cope with these problems, an interesting line of research consists in representing the image with a transform whose building blocks match important signal structures. Unfortunately, the price to pay for such a freedom is that no genuine ONB can be used and a new coding paradigm has to be adopted. In the following, we will basically derive a coding scheme that tries to preserve pre-defined structures in a sequence of frames. More specifically we view such a sequence as a 3-D space-time signal $I(x, y, t)$ and we will try to efficiently encode coherent spatio-temporal structures.

The approach we have chosen relies on expanding the signal as a linear superposition of N generalized waveforms g_γ, tuned to match the requested structures and selected among a vast library $\mathcal{D}$:

$$I = \sum_{i=0}^{N-1} c_i g_{\gamma_i}. \tag{1}$$

The only constraint on the collection $\mathcal{D} = \{g_\gamma, \gamma \in \Gamma\}$ is that it is dense in the space of finite energy signals. In the following, we will refer to g_γ as an atom and to $\mathcal{D}$ as a dictionary. The parameter set Γ can be an anonymous set of labels but it usually carries important information about the atoms, for example space and frequency localization. Of course we also wish that the necessary parameters in this expansion, namely the set of coefficients c_i and indexes γ_i yield good compression performances; this leads to a generic requirement about Eq. (1), namely that this expansion is sparse enough.

Without more constraints on $\mathcal{D}$, and in particular if it is not an ONB, there is generally no unique solution to Eq. (1). One possible solution can be to look for the sparsest exact expansion, that is minimizing the number of coefficients in Eq. (1). This unfortunately leads to a daunting, NP hard, combinatorial optimization problem [14]. A close solution may be provided by relaxing this problem and trying to minimize the ℓ^1 norm of the coefficients which leads to the Basis Pursuit algorithm, deeply studied by Donoho and collaborators [15]. Interestingly, this algorithm sometimes leads to the optimal sparsest solution of Eq. (1) with particular dictionaries [16]–[18]. Alternatively, the Matching Pursuit (MP) algorithm [5] solves Eq. (1) by iteratively decomposing the signal using a greedy strategy. Starting with $R_0 = I$, the n^{th} iteration reads

$$R_n = \langle R_n, g_{\gamma_n} \rangle g_{\gamma_n} + R_{n+1}\,, \tag{2}$$

where the atom g_{γ_n} is the one having maximum correlation with R_n:

$$g_{\gamma_n} = \arg\alpha \max_{\mathcal{D}} |\langle R_n, g_\gamma \rangle|, \tag{3}$$

where α is a positive constant that depends on the search strategy and is equal to 1 in the exhaustive search case. After N steps MP yields a sparse approximation:

$$I = \sum_{i=0}^{N-1} \langle R_i, g_{\gamma_i} \rangle g_{\gamma_i} + R_N, \tag{4}$$

where R_N is the residual error. Matching Pursuit converges [19], that is $\|R_N\| \to 0$ when N tends to infinity and converges even exponentially in finite dimension [5]:

$$\|R_N\|^2 \lesssim (1 - \alpha^2\beta^2)^N \tag{5}$$

where β is a constant that solely depends on $\mathcal{D}$ and is getting close to 1 when the redundancy increases. Recently more constructive results have been obtained concerning the approximation properties of greedy algorithms [18] but their description is beyond the scope of this paper. As already shown in [3], MP is particularly well suited for low rate and adaptive coding of visual information because it easily yields scalable streams by simply truncating Eq. (4). A good approximation is obtained with few well chosen components, mostly because MP will first pick the most prominent signal structures in the dictionary. This property makes it particularly useful at low and medium bit rates.

B. The 3-D Spatio-temporal Overcomplete Basis

The 3-D atoms, which consist of separable spatial and temporal components, should have a structure able to efficiently represent the spatial image content, as well as temporal evolution along motion trajectories. Many approaches have been proposed to improve image representation (e.g. curvelets [1], bandelets [20], contourlets [21]) and all underline that an efficient image representation should have the following properties: (i) multiresolution, (ii) localization: the basis functions should be localized in space and frequency (iii) directionality: the basis functions should be oriented at different directions (iv) anisotropy: the basis functions should have a variety of elongated shapes with different aspect ratios.

As for the spatial part of our dictionary, we use the same construction as proposed in [3], that we sketch here for completeness. Two spatial mother atoms have been proposed, satisfying *the localization* property, a 2-D Gaussian and its 2^{nd} partial derivative,

$$g_1(x, y) = \frac{1}{\sqrt{\pi}} e^{-(x^2+y^2)}, \tag{6}$$

$$g_2(x, y) = \frac{2}{\sqrt{3\pi}} (4x^2 - 2) e^{-(x^2+y^2)}. \tag{7}$$

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

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

- Shift:

$$\mathcal{U}_{(x_0,y_0)} g = g((x - x_0), (y - y_0)), \tag{8}$$

- Orientation:

$$\mathcal{U}_\theta g_2 = g_2(r_{-\theta}(x, y)), \tag{9}$$

- Scaling:

$$\mathcal{U}_a g_1 = \frac{1}{a} g_1(\frac{x}{a}, \frac{y}{a}), \tag{10}$$

$$\mathcal{U}_{(a_1,a_2)} g_2 = \frac{1}{\sqrt{a_1 a_2}} g_2(\frac{x}{a_1}, \frac{y}{a_2}), \tag{11}$$

For implementation issues, spatial position (x_0, y_0) sweeps the whole image and orientation may take 32 values $\theta = \frac{i\pi}{32}$, where $i = 0, \ldots, 31$. The scaling factor a_j, $j = 1, 2$, is logarithmically distributed as $a_j = 2^{\frac{i}{2}}$, with $i = 0, \ldots, 2\lfloor \log(\frac{size}{6}) \rfloor$.

The temporal functions on the other hand should satisfy the following objectives. They should capture most of the signal energy in the low-pass temporal frequencies with few elements, as this will reduce the ghosting artefacts at low bit rates. They should satisfy multiresolution and localization properties in order to encompass the wide range of temporal behavior observed in natural scenes. These properties are achieved by selecting a β-spline $\beta^n(t)$ function [22]. Trading-off between temporal and frequency decay, the order of $\beta^n(t)$ should be $n \geq 2$. While testing n, the 3^{rd} order β-spline $\beta^3(t)$ resulted in good performance for a GOP size of 16. The temporal part of the dictionary is thus generated by shifting and scaling the β-spline

$$\mathcal{T}_{t_0,s} \beta^3 = \beta^3(\frac{(t - t_0)}{s}). \tag{12}$$

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

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

C. Overview of the MP3D Coding Scheme

The building blocks of the novel Matching Pursuit video encoder proposed in this paper, are represented in Figure 1. The MP3D coder consists of two main modules, namely (i) the motion-adaptive 3-D spatio-temporal transform and (ii) the embedded quantization and coding.

The video sequence is first segmented into group of pictures (GOP) of size N, with $N = 16$ in the remaining of this paper. The motion-adaptive 3-D transform performs a motion estimation in the GOP, in order to define the motion fields in each frame. These eventually generate motion trajectories along the successive frames of the GOP. Matching Pursuit is then implemented with a heuristic search algorithm. It provides a sparse representation of the video information in a series of most relevant characteristics, or atoms, that are displaced along the motion trajectories. In a sense, this operation has the same objective as the motion-compensated temporal filtering (MCTF) [11], where the signal is filtered in the temporal dimension along a given trajectory. Finally, the atom parameters are then quantized and progressively encoded to generate a scalable video stream. Lossless coding (DPCM and arithmetic coding) is applied to the motion field parameters, that are sent as a constant rate side information layer to the decoder. The motion adaptive transform and embedded coding stages are described in details in the next sections.

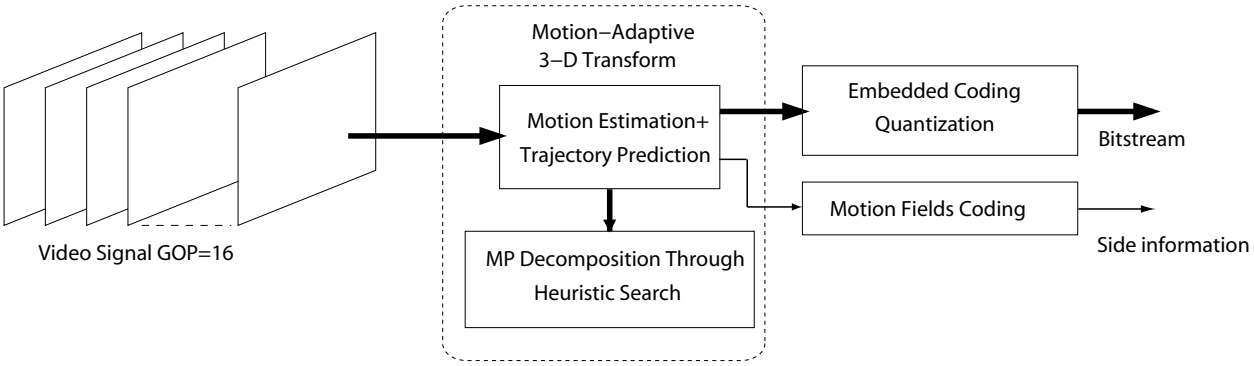


Fig. 1. Block diagram of the Motion-Adaptive Matching Pursuit encoder (MP3D).

IV. The Motion-Adaptive 3-D Transform

A. Motion Estimation and Trajectory Prediction

A key element to efficient video coding consists in efficiently exploiting the temporal redundancy between pictures. This is generally done by motion estimation, which finds the best mapping between successive frames. The motion-adaptive 3D spatio-temporal transform proposed in this work relies on the definition of motion trajectories that correspond to the movement of spatial atoms within the GOP.

The motion trajectories, which could be due to either local or global motion, are computed through the estimation of the motion fields in video frames. Motion vectors, which form motion fields in a frame, are computed here using a block-matching (BM) based technique, in the forward direction. Dividing the frame domain into non-overlapping blocks induces the smoothness of the estimated motion fields. Block matching is illustrated in Figure 2. A block $\mathcal{B}_m$ in the current frame i, is mapped to the best matching block $\mathcal{B}'_m$ in the previous frame $i-1$. Equivalently, the error between $\mathcal{B}_m$, and its parent in the previous frame, is minimized. The spatial displacement vector d_m between these two blocks is the motion vector assigned to each pixel in block $\mathcal{B}_m$. Forward motion vectors are eventually losslessly encoded and transmitted as side information to the decoder.

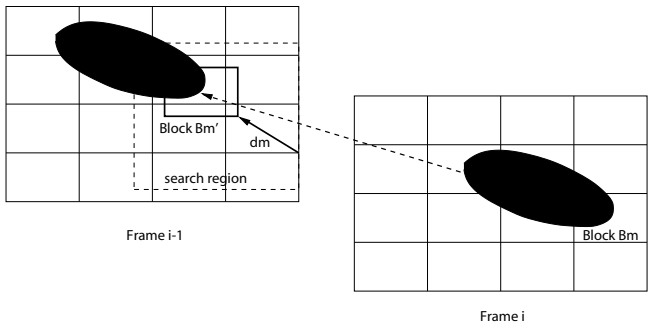


Fig. 2. The block-matching algorithm

The 3-D atom is constructed by propagating its spatial component along the motion trajectory passing by its center in the reference frame, chosen dynamically as the frame with the largest energy in the GOP. A

motion trajectory is defined as the displacement of a pixel, or group of pixels in a frame, toward previous, as well as successive frames. Block matching produces motion fields for each picture of the GOP and pixels uniquely reference parent pixels in the preceding picture. The backward part of the motion trajectory is thus directly given by the motion fields, and follows the motion vectors of the successive blocks toward previous frames in the GOP. However, the motion mapping defined by block matching is unfortunately not bijective, since parent pixels may have several children in the following pictures (see Figure 3). There is no guarantee that two different blocks in frame i do not reference overlapping blocks in the frame $i-1$. Forward motion trajectories could be estimated with a backward motion prediction algorithm (i.e., where blocks in frame i are mapped to blocks in frame $i+1$), but this solution is definitely too expensive in terms of complexity, and coding overhead. Instead, the forward part of the motion trajectories is inferred from the forward predicted motion fields, as follows.

A selection strategy is established a priori, in order to compute the most likely forward trajectory of group of pixels in frame i. A given block $\mathcal{B}_m$ in frame f_i, is mapped to the best matching block ${\mathcal{B}^\star}_m$ in frame f_{i+1} using only the motion vectors derived from forward block matching estimation. The two following criteria allow to select the best trajectory, (i) the minimum distance to the center of the block, and (ii) the scanning order. In the case where more than one motion vector from the frame $i+1$, point to a block that overlaps with block $\mathcal{B}_m$ in frame f_i, the selection is based on the nearest neighbor criteria. In the low probability case where this criteria is not sufficient to choose the best candidate vector, the scanning order is determinant. Note that some blocks in frame f_i may not have any children in the frame $i+1$, which simply means that the motion trajectory ends in frame i. The selection of the motion trajectories

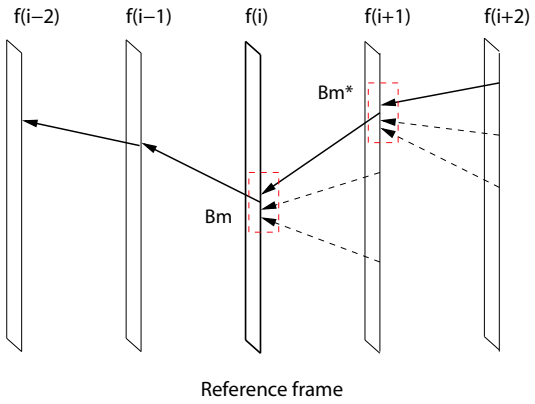


Fig. 3. Example of trajectory prediction, for a block in frame i.

is represented in the remaining of the paper by the generic motion mapping operator $\mathcal{W}$ as,

$$\mathcal{W}_{i\rightarrow i+1}(I(x,y,i)) \approx I(x,y,i+1) \tag{13}$$

where $I(x,y,i)$ denotes the samples of frame i in the video sequence I. Figure 3 illustrates the steps involved during the trajectory prediction. The dotted lines correspond to possible paths which are discarded during trajectory prediction. Thanks to the motion mapping definition, the 3D Matching Pursuit encoder only transmits forward motion vectors to the decoder, and all trajectories are computed similarly at encoder and decoder, according to the definition of $\mathcal{W}$.

B. The Matching Pursuit Expansion

The second part of the 3-D transform consists in computing a decomposition of the GOP in a series of spatio-temporal atoms, by applying the Matching Pursuit algorithm along motion trajectories. A full 3D search is obviously too computationally complex as it requires to evaluate $N \times D$ scalar products, where D is the cardinality of the dictionary and N the number iterations. Previous implementations [2]

that make use of the shift-invariance property of dictionary atoms in Fast Fourier Transform algorithm, are not applicable here due to the motion adaptation step.

Hence, another effective search policy has been designed to trade off the complexity against the approximation performance, and is based on a heuristic search algorithm, described by Algorithm 1. The iterative search algorithm first selects the frame with the highest energy, in the GOP $I^n(x, y, t)$, where $I^0(x, y, t)$ represents the original pictures. It then performs an exhaustive Matching Pursuit search in the selected frame, in order to find the M candidates among the spatial atoms in the dictionary, that best fit the picture characteristics. A fast Matching Pursuit implementation is used, based on a FFT algorithm: it allows computing scalar products with all translated versions of a single atom using one FFT. Each one of the M candidates[1] is then used to build spatio-temporal atoms, aligned on the motion trajectories according to $\mathcal{W}$, and with the temporal functions defined in Section III-B. Each one of these spatio-temporal atoms are compared to the GOP residual signal, $I^n(x, y, t)$, in terms of the energy of the projection coefficients, following Eq. (3). The motion-adaptive spatio-temporal atom that best fits the GOP residual signal is selected. The residual signal is updated accordingly to become $I^{n+1}(x, y, t)$. The process is repeated until the signal expansion is long enough, or until a residual error energy threshold has been reached. Note that even if the Matching Pursuit does not perform a full search, the algorithm still converges quite rapidly, i.e., the constant α from Eq. (5) stays close to 1.

Algorithm 1 The Heuristic Search Algorithm.

1: Let $I(x, y, t), t = 1..N$ be a block of frames
2: Select a reference frame r with the largest energy
3: Use the 2D exhaustive search FFT-based algorithm to find the best M uncorrelated candidates among spatial atoms
4: Search for the best mapped 3-D atom starting from the M candidates
5: Update the residual $I^n(x, y, t)$ accordingly and iteratively get back to step 1.

V. Embedded Coding and Quantization

A. Scalable coding

After the motion-adaptive 3D transform, coefficients and atoms have to be scalably encoded in order to provide a rate and geometry-adaptive video signal representation. This stage is key to fully benefit of the intrinsic scalability properties of Matching Pursuit expansions over a dictionary built on geometric laws. Each spatio-temporal atom index needs to be coded and transmitted along with its coefficient. The atom index is represented by the tuple $(p_x, p_y, p_t, a_x, a_y, a_t, \theta)$, that costs an average of ρ bits, where (p_x, p_y, p_t), (a_x, a_y, a_t) and θ, respectively represent the position of the atom, the scale and the spatial rotation parameters.

Since Matching Pursuit by nature produces a progressive series of atoms, where most energetic components appear first, it makes sense to code atoms in order of their appearance. The stream then offers a great flexibility, since even a simple truncation ensures that the most important features are preserved. Coefficients can be coded by taking into account the exponential decay of their magnitude, as proposed in [23]. An approach based on successive refinement of information [24] can also be implemented, since the statistics of the exponential distribution of coefficient magnitude are known. Both approaches allow to reduce the coding rate of the coefficients, and provides a fully progressive stream. However, no significant gain can be obtained on the atom indexes, since their order is mostly random. Hence, they both work well when atom index size ρ is small. However, the index size may be as large as a few tens of bits for 3D atoms.

[1] M is chosen to be proportional to the number of blocks in a picture, in the current implementation.

An alternative coding method consists in trying to reduce the index coding rate by changing the initial order of the atoms, possibly at the price of a higher coefficient coding rate. Interestingly, the spatial position parameters p_x and p_y are responsible for almost half of the index coding rate. A natural approach consists in sorting atoms along their spatial position, row- and column-wise, similar to the method proposed in [25]. Run-length coding of spatial position parameters, and lossless arithmetic coding of the remaining parameters could then save up to 7 bits on the average index size (in the case of CIF video format). Nevertheless, in such a scheme, the stream is no longer progressive, since atoms are not sorted along their magnitude anymore, but rather according to their spatial positions. Trying to get the best out of

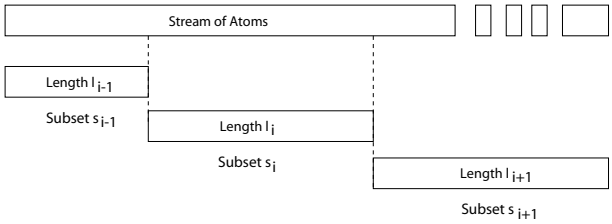


Fig. 4. The series of atoms is divided into energy subsets, as a compromise between coding efficiency and flexibility.

both coding alternatives, the embedded coding in the MP3D encoder initially divides the series of atoms in S disjoint sub-sets s_i, where each subset contains l_i elements as shown in Figure 4. These subsets can be seen as energy sub-bands. Their number is dictated by scalability requirements (i.e., the number of target decoding rates), and represents a tradeoff between stream flexibility, and coding efficiency, that respectively increases and decreases with S. In each subset, atoms are sorted according to their spatial positions, that are further run-length encoded. Other index parameters and quantized coefficients are encoded with a context adaptive arithmetic encoder [26]. The resulting bitstream is now piecewise progressive, and optimal truncation points can be set at subset limits. The rate control problem belongs to a general class of bit allocation problems under multiple rate constraints, and is discussed in details below.

B. Coefficient Quantization

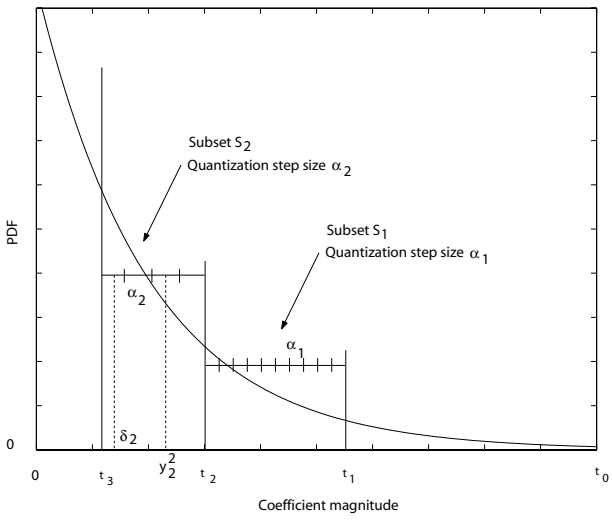


Fig. 5. Example of the construction of energy sub-bands s_i

As already observed in [25], the distribution of MP coefficients magnitudes obeys an exponential pdf:

$$f(x) = \mu^{-1} e^{-\frac{x}{\mu}}, x > 0, \mu > 0. \tag{14}$$

Without constraint on the number of quantization levels, the mean-square error optimal Entropy Constrained Scalar Quantizer (ECSQ) for the exponential pdf is the uniform scalar quantizer designed in [27]. We choose also to use a uniform quantizer in each energy subband, allowing however for a different step size in each coefficient sub-set. This allows us to conveniently model rate and distortion and develop an efficient rate allocation scheme. As introduced before, the subset s_i contains l_i coefficients, whose magnitudes belong to the interval $[t_{i+1}, t_i[$ (see Figure 5). With these notations, s_0 gathers the highest energy atoms, and we assumed $t_0 \to \infty$. Under such assumptions, the number of coefficients in subset s_i is given by $l_i = L(\exp(-\frac{t_{i+1}}{\mu}) - \exp(-\frac{t_i}{\mu}))$, where L is the total number of atoms. Uniform quantization in the subset s_i then generates reconstruction centroids y_k^i that are given by

$$y_k^i(\alpha_i, t_{i+1}, \mu) = k\alpha_i + t_{i+1} + \delta_i, \tag{15}$$

where k represents the quantization bin, and α_i is the quantization step size. Following the notation used in [27], the reconstruction offset in each set is given by:

$$\delta_i = \mu - \frac{\alpha_i e^{-\frac{\alpha_i}{\mu}}}{1 - e^{-\frac{\alpha_i}{\mu}}}. \tag{16}$$

Using conditional probabilities, the entropy H_i in each subset can be written as:

$$\begin{aligned} H_i = \frac{P_0}{\log 2} \Bigg[& \frac{\frac{\alpha_i}{\mu} e^{-\frac{\alpha_i}{\mu}} \left(1 - e^{(-\frac{t_i - t_{i+1} - \alpha_i}{\mu})}\right)}{(1 - e^{(-\frac{\alpha_i}{\mu})})^2} \\ & - \frac{\log P_0 + (\frac{t_i - t_{i+1} - \alpha_i}{\mu} - \log P_0) e^{(-\frac{(t_i - t_{i+1})}{\mu})}}{1 - e^{(-\frac{\alpha_i}{\mu})}} \Bigg], \end{aligned} \tag{17}$$

with P_0 equal to

$$P_0 = \frac{1 - e^{-\frac{\alpha_i}{\mu}}}{1 - e^{-\frac{t_i - t_{i+1}}{\mu}}}. \tag{18}$$

Under the assumption that modern coding methods such as arithmetic coding [26] can achieve rates close to the entropy, the rate R_i required to code the atoms of the subset s_i is given by $R_i = l_i\ (H_i + \rho)$, where ρ is the average number of bits to code the atom index, which does not depend on s_i. The mean-square distortion D_i in each subset s_i is finally composed of two terms: the quantization error in s_i, $\Delta_q^i(\alpha_i)$, and the distortion $\Delta(t_{i+1})$ due to discarding the coefficients with magnitude smaller than t_{i+1} (see Figure 5). Equivalently,

$$D_i = \Delta_q^i(\alpha_i) + \Delta(t_{i+1}), \tag{19}$$

where

$$\Delta_q^i(\alpha_i) = \left(e^{-\frac{t_{i+1}}{\mu}} - e^{-\frac{t_i}{\mu}}\right) \left(\mu^2 - \frac{\alpha_i^2 e^{-\frac{\alpha_i}{\mu}}}{(1 - e^{-\frac{\alpha_i}{\mu}})^2}\right), \tag{20}$$

and

$$\Delta(t_{i+1}) = 2\mu^2(1 - e^{-\frac{t_{i+1}}{\mu}}) - t_{i+1} e^{-\frac{t_{i+1}}{\mu}}(t_{i+1} + 2). \tag{21}$$

Details about the computation of H_i and D_i are given in Appendix I.

C. Rate Allocation

For scalability issues, the bitstream should be encoded efficiently at several target rates, that can be directly defined by the application. The rate allocation becomes an optimization problem with multiple constraints. The bit-stream has to be efficiently decoded at target rates $\{r_0, r_1, \ldots, r_{N-1}\}$, and the corresponding distortions $\{d_0, d_1, \ldots, d_{N-1}\}$ should be as close as possible to the best distortion achievable at these target rates. This optimization problem can be formalized with Lagrange multipliers $\vec{\lambda}$, and becomes equivalent to minimizing the cost function $J(\vec{\lambda}) = \mathcal{D} + \vec{\lambda}^T \vec{\mathcal{R}}$ subject to $\mathcal{R}_i \leq r_i$ for $i = 0, \cdots, N-1$, where $D = Tr\left([\Delta_q^0, \ldots, \Delta_q^{N-1}] \cdot U_1\right) + \sum_{i=0}^{N-1} \Delta(t_{i+1})$, $\vec{\mathcal{R}} = [R_0, R_1, \ldots, R_{N-1}] \cdot U_1$, and U_1 is an upper triangular matrix of ones. The constrained optimization problem now consists in finding the thresholds, t_i, and the quantizer step sizes, α_i, that minimize $J(\vec{\lambda})$.

Instead of solving the complex global optimization problem, the proposed rate allocation strategy adopts a greedy approach, that optimizes the quantization in each independent subset successively. Since atoms are allocated to subsets according to their magnitude, subsets are naturally assigned different priority levels, with s_0 becoming the most important one. The rate allocation algorithms starts by minimizing $J_0(\lambda_0) = D_0 + \lambda_0 R_0$ under the rate constraint $R_0 \leq r_0$ to find either the threshold t_1 (or equivalently l_0) and the step size α_0. It then optimizes iteratively $J_i(\lambda_i) = D_i + \lambda_i R_i$ under the rate constraint $R_i \leq (r_i - \sum_{k=0}^{i-1} R_k)$. The rate allocation is summarized in Algorithm 2.

Algorithm 2 The Rate Allocation Algorithm

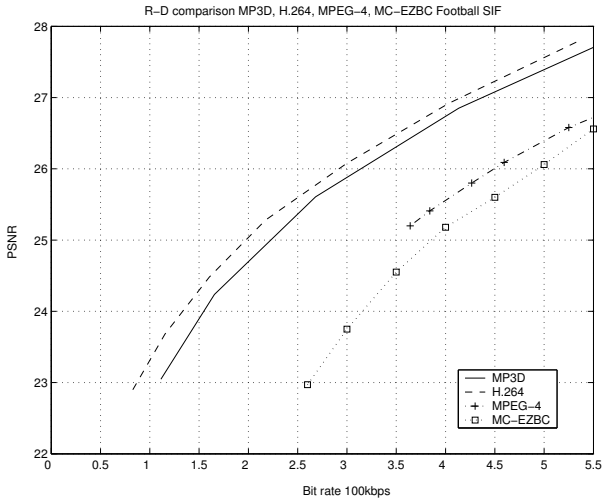
Let $\{r_0, r_1, \ldots, r_{N-1}\}$ be the multiple rate constraints.
Set the rightmost threshold $t_0 \rightarrow \infty$
Find $(\alpha_0, t_1) = \arg\ \min(J_0(\lambda_0) = D_0 + \lambda_0 R_0)$ subject to $R_0 \leq r_0$
for $i = 1 \ldots N-1$ **do**
  Find $(\alpha_i, t_{i+1}) = \arg\ \min(J_i(\lambda_i) = D_i + \lambda_i R_i)$ subject to $R_i \leq (r_i - \sum_{k=0}^{i-1} R_k)$
end for

Various approaches can be used to minimize each individual cost function $J_i(\lambda_i)$. Heuristic approaches based on the pruning strategy operating on trees have been proven to be efficient [28], [29]. A tree of Q subtrees can be built for each subset s_i, where each subtree is a unitree of $Card(\mathcal{A})$ nodes, with $\mathcal{A}$ the set of possible quantization step sizes α. A set $\mathcal{A}$ of fifteen values has been verified to work well over a large range of bit rates, and therefore we set $\mathcal{A} = \{120, 100, 80, 70, 60, 50, 40, 35, 30, \cdots, 10, 5\}$. Each node of a unitree corresponds to a different rate-distortion tuple (R_i, D_i), and the leaf is the lowest distortion node (respectively the node with the highest rate). Each unitree then corresponds to a different number of atoms l_i in the subset (i.e., a different value of t_{i+1}), and the number of unitrees Q is an arbitrary parameter that is related to the search space dimension. The search can be limited to values of l_i that are close to $L_i = \frac{(r_i - \sum_{k=0}^{i-1} R_k)}{\rho}$, the maximum number of atoms permitted under the rate constraints, since the index size is generally larger than the coefficient entropy. In the current implementation, Q is set to 10, and l_i is uniformly distributed between $0.75L_i$ and L_i. Finally, the minimization of $J_i(\lambda_i)$ consists in pruning the unitrees for the nodes that violates the rate constraints $R_i \leq (r_i - \sum_{k=0}^{i-1} R_k)$, and then chooses among the Q unitrees, the leaf with the minimal distortion.

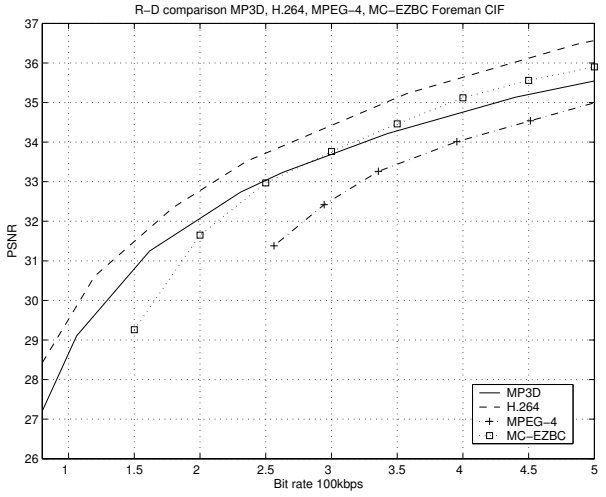
VI. Experimental Results

A. Rate Distortion Comparison Against H.264 and MPEG

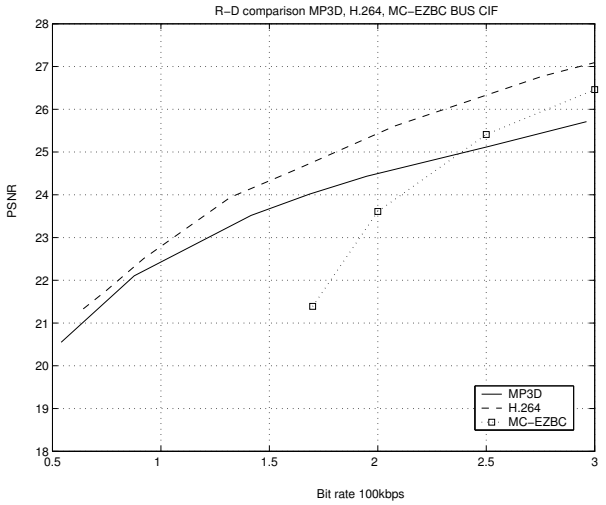
In this section, we evaluate the rate-distortion performances of our codec by comparing it with two reference schemes: MPEG4 [30] and H.264 [31]. The standard Foreman, Football and Bus sequences in CIF format at 30 fps were used to generate the results. In all experiments, we used a GOP size of 16. It can be seen on Fig. 6 (a) and (b) that the PSNR of MP3D is higher than that of MPEG-4 by about 1-1.5 dB for Foreman and Football sequences and over a wide range of bit-rates. Meanwhile, it is



(a) Football sequence



(b) Foreman sequence



(c) Bus sequence

Fig. 6. Rate-distortion performances of MP3D.

only slightly inferior to the performance of H.264, staying within a 1 dB gap (see Fig. 6).We noted that the results for the Foreman sequence are always penalizing our scheme at high rate. On the other hand, our scheme performs better on the Football sequence for example, where it stays close to H.264 over the whole range of bit rates under consideration. This might seem strange since the Foreman sequence contains a lot of important edges. Our codec does not perform very well on textures (see Fig. 7): once most geometrical information has been encoded, PSNR tends to saturate at higher rates. Finally, one should remember that both H.264 and MPEG-4 are non-scalable video coding schemes optimized for compression performance, contrarily to MP3D. In the next section, we compare MP3D to another highly scalable scheme for completeness.

B. R-D Comparison Against MC-EZBC

An objective evaluation of MP3D against MC-EZBC [32], in terms of PSNR performances, is given in Fig. 6. Both schemes have the same GOP size and a single layer coding for MVs. For the Football sequence, MP3D has a higher PSNR than MC-EZBC over the whole studied range of bit rates. For the Foreman and Bus sequences however, there exists a cross-over point where MP3D loses its advantage. At some higher rate, coding the atoms results in a less significant PSNR improvement. Nevertheless, we observed that MP3D is always more efficient at low and medium rate (less than 500 kbps).

C. Visual Quality Comparison

Fig. 7 shows visual comparisons of the first frame from the Football sequence decoded at 550 kbps, using the schemes mentioned before. One can see that H.264 produces more uniform regions. The regions in MP3D are also very smooth, but most prominent edges are well captured due to the nature of the dictionary we used. On the other hand, MP3D lost most textures. The frame coded by MC-EZBC shows a trade-off in representing smooth areas and texture components, while MPEG-4 produces an overall slightly inferior visual quality. Of course these tests are not conclusive, but they allow to show the behavior of MP3D in capturing first the geometrical features in image sequences.

VII. Scalability Properties

The scalability features are intrinsic in MP3D due to the multiresolution structure of the dictionary, the nature of MP and the embedded coding. All these parameters make the bitstream highly scalable, offering 3-D geometric (i.e. spatio-temporal) and SNR scalability. The geometric properties of the dictionary ensure very easy sequence adaptation prior to decoding. As a result, a single bitstream can be decoded at any spatial resolution (as long as the re-scaling is isotropic) and at various frame rates, without resorting to costly re-encoding or post-processing operations. These properties significantly differ from simple transcoding schemes, and we chose to refer to them as geometric scalability.

For example, a coded video signal I of spatial size $W \times H$ with a frame rate F can be spatially decoded into a video signal $\tilde{I}$ of spatial resolution $\alpha W \times \alpha H$ at the same frame rate as follows. First the full atom trajectory is reconstructed at the initial size using the motion field operator $\mathcal{W}$. Then each individual atom is analytically re-scaled by simply transcoding its index values (scales and positions) as described in [2]. The new signal reads :

$$\tilde{I} = \sum_{i=0}^{N-1} \alpha \; c_i \widetilde{\mathcal{W}(g_{\gamma_i})}, \tag{22}$$

where c_i are the atom coefficients and $\widetilde{\mathcal{W}(g_{\gamma_i})}$ corresponds to the motion-mapped atom $\mathcal{W}(g_{\gamma_i})$ after transcoding. We noted that, when transcoding by $\alpha < 1$, thus to a lower resolution, possible aliasing from very small atoms saturate quickly PSNR quality as rate increases. The smallest atoms are simply discarding by stream truncation. Figure 8 shows frame 1 of the Foreman sequence decoded in QCIF format from the bitstream corresponding to CIF format. One sees that spatial resolution adaptation nicely

(a) Original

(b) MP3D

(c) MC-EZBC

(d) H.26L

(e) MPEG-4

Fig. 7. Visual Comparison for Frame 1 of Football decoded at 550 kbps

preserves edges after transcoding. These structures are indeed well captured by our dictionary and the corresponding atoms are simply re-scaled, when decoded at a different resolution. This clearly brings a great advantage in visual quality.

Besides geometric scalability, MP3D provides natural SNR scalability because of the exponential decay of MP coefficients and the embedded quantization. It was noticed in Section V that the amplitude of atom

Fig. 8. Frame 1 of Foreman decoded in QCIF from the CIF bitstream.

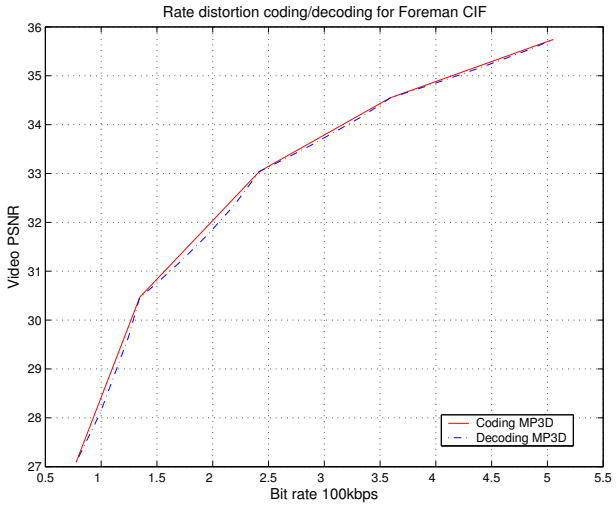


Fig. 9. PSNR of the foreman sequence encoded according to the multi-rate constraint {75, 135, 245, 360, 500} kbps and decoded at various intermediate rates

coefficients has a small variation within most of the subsets (except for the first one). Therefore, simple truncation of the embedded bitstream in a given subset still ensures that the decoder receives most of the signal energy for the available bandwidth as shown on Fig. 9. Finally, Fig. 10 shows frame 1 of the Foreman sequence decoded at 320 kbps from a bitsream, that was encoded using the multiple rate constraints $\{75, 135, 245, 360, 500\}$ kbps, with an average PSNR of 33.8 db.

VIII. Conclusions

In this paper we introduced a video coding scheme based on motion-adaptive decompositions in a redundant dictionary of waveforms. The overcomplete dictionary is designed to model image primitives, mostly edges, that are likely to display coherent trajectories over time. The Matching Pursuit algorithm is first used to compute a compact signal representation. In parallel, the sequence motion field is estimated by classical block matching techniques and used to recover the trajectories of most prominent image primitives. The data is then filtered along these trajectories using a redundant temporal dictionary. An embedded multi-rate allocation method was designed to offer a progressively refinable bit-stream. Target rate points are defined at the encoder, and the decoder can recover those R-D points by truncating the stream. Sub-optimal decoding is still possible in-between pre-defined target rates and simulations show only a slight degradation in R-D performance. The compressed video sequence can further be decoded at any spatial resolution due to the parametric structure of the redundant libraries used to represent the information. These geometric stream manipulations are lightweight and can be performed at the decoder or by some simple network intelligence. Comparisons with state-of-the-art scalable and non-scalable codecs illustrate the good performance of the proposed technique at low bit-rates and motivate its possible use as a base layer in a more general scalable framework. The computational complexity of our scheme is clearly one of its main drawbacks, and faster implementations are currently under study.

Fig. 10. Frame 1 of sequence Foreman decoded at 320 kbps, from the 500kpbs bitstream

APPENDIX I
ENTROPY AND DISTORTION IN COEFFICIENT SUBSETS

Let s_i be the subset of all possible atoms whose coefficient modulus belongs to $[t_{i+1}, t_i]$ according to the exponential distribution in Eq. 14. By assuming a uniform quantization step size of α, the total number of possible quantization levels (or the number of bins) n should be

$$n = \frac{t_i - t_{i+1}}{\alpha}. \tag{23}$$

The conditional probability of bin k given s_i is $p(k|s_i)$ (which is denoted by P_k) is

$$\begin{aligned} p(k|s_i) &= \frac{p(k)}{p(s_i)} && (24)\\ &= \frac{e^{-\frac{t_{i+1}+\alpha k}{\mu}} - e^{-\frac{t_{i+1}+\alpha(k+1)}{\mu}}}{e^{-\frac{t_{i+1}}{\mu}} - e^{-\frac{t_i}{\mu}}} && (25)\\ &= \left(\frac{1 - e^{-\frac{\alpha}{\mu}}}{1 - e^{-\frac{t_i - t_{i+1}}{\mu}}}\right) e^{-\frac{\alpha k}{\mu}} && (26)\\ &= P_0 e^{-\frac{\alpha k}{\mu}}. && (27) \end{aligned}$$

with P_0 is defined in Eq. (18).

Now, the resulting entropy necessary to code all the quantized values in s_i is

$$\begin{aligned} H_i &= -\sum_{k=0}^{n-1} P_k \log_2 P_k \\ &= -\sum_{k=0}^{n-1} P_0 e^{-\frac{\alpha k}{\mu}} \log_2 (P_0 e^{-\frac{\alpha k}{\mu}}) \\ &= \frac{P_0}{\log 2} \sum_{k=0}^{n-1} e^{-\frac{\alpha k}{\mu}} (\frac{\alpha k}{\mu} - \log P_0). \end{aligned}$$

By using the closed form of a finite sum of an arithmetico-geometric progression, we obtain

$$H_i = \frac{P_0}{\log 2}\left[\frac{\frac{\alpha}{\mu}e^{-\frac{\alpha}{\mu}}\left(1-e^{-\frac{\alpha(n-1)}{\mu}}\right)}{(1-e^{-\frac{\alpha}{\mu}})^2} - \frac{\log P_0 + (\frac{\alpha(n-1)}{\mu} - \log P_0)e^{-\frac{\alpha n}{\mu}}}{1-e^{-\frac{\alpha}{\mu}}}\right], \tag{28}$$

Now substituting n of Eq. (23) in Eq. (28), we get the final expression of H_i as in Eq. (17).

When the mean-square error (MSE) measure is used, the reconstruction centroids are defined as:

$$\begin{aligned} y_k^i(\alpha, t_{i+1}, \mu) &= \frac{\int_{t_{i+1}+\alpha k}^{t_{i+1}+\alpha(k+1)} x f_e(x)\,dx}{\int_{t_{i+1}+\alpha k}^{t_{i+1}+\alpha(k+1)} f_e(x)\,dx} \\ &= \alpha k + t_{i+1} + \frac{\int_0^\alpha x f_e(x)\,dx}{\int_0^\alpha f_e(x)\,dx} \\ &= \alpha k + t_{i+1} + \mu - \frac{\alpha e^{-\frac{\alpha}{\mu}}}{1-e^{-\frac{\alpha}{\mu}}} \\ &= \alpha k + t_{i+1} + \delta_i, \end{aligned}$$

where δ_i is defined as in Eq. (16).

Finally, the distortion D_i is defined as the total distortion for coefficients whose magnitude is smaller than t_i which is composed of two terms, one Δ_q^i due to quantization error in s_i and the other $\Delta(t_{i+1})$ due to discarding the coefficients whose magnitude is smaller than t_{t+1}, i.e. $D_i = \Delta_q^i + \Delta(t_{i+1})$.

Δ_q^i is defined as the average cumulative distortion by using the reconstruction codewords y_k^i over the entire subset s_i, i.e.,

$$\Delta_q^i = \sum_{k=0}^{n-1}\int_{t_{i+1}+\alpha k}^{t_{i+1}+\alpha(k+1)} (x-y_k^i)^2 f_e(x)\,dx \tag{29}$$

$$= e^{-\frac{t_{i+1}}{\mu}}\sum_{k=0}^{n-1} e^{-\frac{\alpha k}{\mu}} \underbrace{\int_0^\alpha (x-\delta_i)^2 f_e(x)\,dx}_{\gamma}, \tag{30}$$

Clearly γ does not depend on k and it is given as

$$\gamma = \mu^2(1-e^{-\frac{\alpha}{\mu}}) - \frac{\alpha^2 e^{-\frac{\alpha}{\mu}}}{1-e^{-\frac{\alpha}{\mu}}}.$$

Now, by substituting γ and n in Eq. (29) we obtain Δ_q^i as

$$\Delta_q^i = (e^{-\frac{t_{i+1}}{\mu}} - e^{-\frac{t_i}{\mu}})\left(\mu^2 - \frac{\alpha^2 e^{-\frac{\alpha}{\mu}}}{(1-e^{-\frac{\alpha}{\mu}})^2}\right).$$

The second term $\Delta(t_{i+1})$ is a defined as follows

$$\begin{aligned} \Delta(t_{i+1}) &= \int_0^{t_{i+1}} x^2 f_e(x)\,dx \\ &= e^{-\frac{x}{\mu}}(x^2+2x\mu+2\mu^2)\Big|_{x=t_{i+1}}^{x=0} \\ &= 2\mu^2(1-e^{-\frac{t_{i+1}}{\mu}}) - t_{i+1}e^{-\frac{t_{i+1}}{\mu}}(t_{i+1}+2), \end{aligned}$$

which gives the final expression of D_i as in equation (19).

REFERENCES

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

[2] A. Rahmoune, P. Vandergheynst, and P. Frossard, "MP3D: A highly scalable video coding scheme based on matching pursuit," in *Proceeding IEEE ICASSP*, vol. 3, Montreal, May 2004, pp. 133–136.

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

[4] P. Frossard, "Robust and multiresolution video delivery : From h.26x to matching pursuit based technologies," Ph.D. dissertation, Swiss Federal Institute of Technology Lausanne, 2000.

[5] 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.

[6] G. Karlsson and M. Vetterli, "Three-dimensional subband coding of video," in *Proceeding IEEE ICASSP*, vol. 2, Apr 1988, pp. 1100–1103.

[7] J.-R. Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Transactions on image processing*, vol. 3, pp. 559–571, Sept 1994.

[8] D. Taubman and A. Zakhor, "Multirate 3-d subband coding of video," *IEEE Transactions on image processing*, vol. 3, no. 5, pp. 572–588, Sept 1994.

[9] B.-J. Kim and W. A. Pearlman, "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (3d-spiht)," in *IEEE Data compression conference*, March 1997, pp. 251–259.

[10] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on circuit and systems for video technology*, vol. 6, pp. 234–250, June 1996.

[11] S.-J. Choi and J. W. Woods, "Motion-compensated 3-d subband coding of video," *IEEE Transactions on image processing*, vol. 8, no. 2, pp. 155–167, Feb 1999.

[12] A. Secker and D. Taubman, "Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression," *IEEE transactions on image processing*, vol. 12, no. 12, december 2003.

[13] B. Pesquet-Popescu and V. Bottreau, "Three-dimensional lifting schemes for motion compensated video compresssion," in *IEEE Proc.Int. Conf. Acoust. Speech. and Sig. Pro.*, May 2001, pp. 1793–1796.

[14] Davis G., Mallat S. and Avellaneda M., "Adaptative greedy approximations," *Constructive Approximations*, 1997, springer-Verlag New YORK INC.

[15] S. Chen and D. Donoho, "Atomic decomposition by basis pursuit," in *SPIE International Conference on Wavelets*, San Diego, July 1995.

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

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

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

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

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

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

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

[23] P. Frossard, P. Vandergheynst, R. F. i Ventura, and M. Kunt, "A posteriori quantization of progressive matching pursuit streams," *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp. 525–535, February 2004.

[24] W. H. R. Equitz and T. M. Cover, "Successive refinement of information," *IEEE Transactions on Information Theory*, vol. 37, no. 2, pp. 269–275, March 1991.

[25] R. Neff and A. Zakhor, "Very low bit-rate video coding based on matching pursuits," *EEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 158–171, February 1997.

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

[27] G. J. Sullivan, "Efficient scalar quantization of exponential and laplacian random variables," *IEEE Transactions on information theory*, vol. 42, no. 5, pp. 1365–1374, Sept 1996.

[28] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling," *IEEE Transactions on Information Theory*, vol. 35, pp. 299–315, Mar 1989.

[29] E. A. Riskin, "Optimal bit allocation via the generalized BFOS algorithm," *IEEE Transactions on Information Theory*, vol. 37, pp. 400–402, Mar 1991.

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

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

[32] "MC-EZBC software, http://mpeg.nist.gov/cvsweb/."