New Dictionaries for Matching Pursuits Video Coding

C. De Vleeschouwer and B. Macq

Laboratoire de Télécommunications et Télédétection
Université catholique de Louvain
B-1348 Louvain-la-Neuve
Tel. : +32 10 47.41.05 - Fax : +32 10 47.20.89
E-mail : devlees@tele.ucl.ac.be

Abstract

"Matching Pursuits" is a signal expansion technique whose efficiency for video coding has already been demonstrated in the MPEG-4 community. In this paper, we foccus our attention on some aspects of the algorithm. Firstly, the relevance of the orthogonalization of the signal expansion is investigated. Secondly, a subband dictionary is proposed as an alternative to the Gabor dictionary, used up to now. Equivalent levels of quality are achieved with both dictionaries but the computational cost is significantly reduced when using the subband one. Eventually, the entropy coding stage of the algorithm is studied.

1 Introduction

Efficient video coders use motion models. In general, a future frame is predicted from the past frame, using local motion information. Then, the prediction error, also named Displaced Frame Difference (DFD), is compressed using techniques involving some signal expansion methods. Typically, the Discrete Cosine Transform is used but this paper focuses on a new emerging DFD compression technique. The method is based on the Matching Pursuits [1] that decomposes a signal into a linear expansion of waveforms selected from a redundant dictionary of functions. This is a complete expansion, i.e. any function can be written as a linear combination of the dictionary reference functions.

Let's remind the principle of Matching Pursuits (MP). Given a large and redundant dictionary $\mathcal{D} = (g_\gamma)_{\gamma\in\Gamma}$ of vectors in $L^2(R)$, such that $\|g_\gamma\| = 1$, MP performs an adaptive expansion of any vectors f in $L^2(R)$ over a set of waveforms selected from $\mathcal{D}$, in order to best match its structures. Let $g_{\gamma_0} \in \mathcal{D}$. The vector f can be decomposed into

$$f = < f, g_{\gamma_0} > g_{\gamma_0} + Rf \qquad (1)$$

where Rf is the residual vector after approximating f in the direction of g_{γ_0}. Clearly, g_{γ_0} is orthogonal to Rf, hence

$$\|f\|^2 = | < f, g_{\gamma_0} > |^2 + \|Rf\|^2 \qquad (2)$$

To minimize $\|Rf\|$, we must choose g_{γ_0} such that $| < f, g_{\gamma_0} > |$ is maximum. This is the first step of the approximation procedure. It is repeated iteratively on the obtained residue. If the decomposition is carried up to order m, f is decomposed into a sum of m **atoms** $< R^n f, g_{\gamma_n} > g_{\gamma_n}$ and of the mth order residue $R^m f$, i.e. :

$$f = \sum_{n=0}^{m-1} < R^n f, g_{\gamma_n} > g_{\gamma_n} + R^m f \qquad (3)$$

Convergence of the process is ensured by equation (2), i.e. the residue energy is always decreasing. MP can be viewed as a way to build signal adapted bases. By progressively isolating the structures of the signal that are coherent with respect to the chosen dictionary It provides an adaptive signal representation in which the most significant coefficients are first extracted. This is a key issue for very low bit rate coding applications. Moreover, in the context of prediction error coding, this selective transmission of significant coefficients is even more relevant as the DFD is a sparse image, containing packets of information where the motion prediction model is inadequate.

The extension of the MP technique to the DFD coding has already successfully been implemented by Neff and Zakhor [2] and Banham [3]. Firstly, a dictionary composed of a set of 2-D separable Gabor functions is chosen. Secondly, the dictionary is extended to the picture domain by allowing the centre of each function to be translated in each pixel position. A direct extension of the MP algorithm would require to examine each 2-D dictionary structure at all possible pixel location in the DFD. As stated by Neff and Zakhor [2], assuming that the DFD is sparse with packets of energy where the motion prediction is inadequate, we can limit the search in a window around these high-energy packets.

In section 2, an original method performing the orthogonalization of the signal expansion is described. Its relevance for MP encoding is discussed. In section 3, a new subband dictionary is proposed to reduce the signal expansion process complexity by exploiting results achieved in wavelet and multiresolution the-

0-8186-8821-1/98 $10.00

ory. In section 4, an original method is proposed to code the location of the expended signal coefficients. It brings some improvements to the one proposed by Neff and Zakhor [2] by taking into account some a priori knowledge deduced both from the motion information and from the predicted frame.

2 "In the Loop" Back Projection

During the expansion process, after m steps, the energy of the residue could be reduced if f was approximated by its orthogonal projection over the space generated by $(g_{\gamma_n})_{0\leq n<m}$. This step, called back-projection, is only useful when the number of atoms reaches the size of the original signal [1]. Let's examine this process in detail. Mallat [1] performed it when the complete set of vectors $(g_{\gamma_n})_{0\leq n<m}$ had been chosen. This required the computation of $< R^m f, g_{\gamma_n} > \forall 0 \leq n < m$. Here, the back projection, when necessary, is performed at each step of the algorithm. So, it is referred to as an **"in the loop" back projection.** Let V_m be the space generated by $(g_{\gamma_n})_{0\leq n<m}$ and W_m be the orthogonal complement of V_m in $L^2(R)$. P_S is the orthogonal projector on the subspace S, i.e. for any f in $L^2(R)$, $P_S f$ is the closest vector to f that can be written as a linear expansion of basis vectors of S. After $m-1$ steps, f is decomposed into :

$$f = P_{V_{m-1}} f + P_{W_{m-1}} f = \sum_{n=0}^{m-2} c_{n,m-1} g_{\gamma_n} + P_{W_{m-1}} f \tag{4}$$

with $c_{n,m-1} =< P_{W_n} f, g_{\gamma_n} > + \sum_{k=n+2}^{m-2} x_{n,k}$, where $x_{n,k}$ results from the back-projection of $R^k f$ on g_{γ_n}. To explicit the way it is computed, lets examine what happens at step m. First, $g_{\gamma_{m-1}}$ is chosen to maximize $| < P_{W_{m-1}} f, g_{\gamma_{m-1}} > |$. f is decomposed into :

$$f = P_{V_{m-1}} f + < P_{W_{m-1}} f, g_{\gamma_{m-1}} > g_{\gamma_{m-1}} + R^m f \tag{5}$$

and

$$P_{W_{m-1}} f =< P_{W_{m-1}} f, g_{\gamma_{m-1}} > g_{\gamma_{m-1}} + R^m f \tag{6}$$

The back projection step consists to the orthogonal projection of $R^m f$ on V_m. So, as $R^m f$ is orthogonal to $g_{\gamma_{m-1}}$,

$$R^m f = \sum_{j=0}^{m-2} x_{j,m} g_{\gamma_j} + P_{W_m}(R^m f) \tag{7}$$

At this step, $P_{W_m}(R^m f) = P_{W_m} f$. The knowledge of coefficients $x_{j,m}$, $0 \leq j < m-1$ permits to loop the loop and to go on the iterative process from equation (4). Inserting equation (7) in equation (6), the orthogonality of $P_{W_{m-1}} f$ with each of the $m-1$ vectors of $(g_{\gamma_i})_{0\leq i<m-1}$ provides a system of $m-1$ equations with $m-1$ unknowns, i.e. $\forall\ 0 \leq i < m-1$,

$$< P_{W_{m-1}} f, g_{\gamma_i} >= 0 \tag{8}$$

This can be written in the form of a $(m-1)\times(m-1)$ linear systems $Ax = b$ with $a_{i,j} =< g_{\gamma_i}, g_{\gamma_j} >$ and $b_i = - < P_{W_{m-1}} f, g_{\gamma_{m-1}} >< g_{\gamma_{m-1}}, g_{\gamma_i} >$. At each step, only the coefficients b_i, i.e. values of $< g_{\gamma_{m-1}}, g_{\gamma_i} >$, have to be computed. These values are used to fulfil the last row and last column of matrice A for the next step. Actually, for MP applications, one may store the correlation value of each couple of dictionary functions, the shift between these functions being a parameter. Only shift values resulting in overlapped functions have to be stored. For a couple of function (f_1, f_2) of length l_1, l_2, $l_1 + l_2 - 1$ values have to be stored. For 2-D applications, the use of separable dictionary functions only requires the storage of the correlation values for the 1-D functions. This shows that the cost of the computation of the coefficients of equations (8) is extremely low.

The resulting system is a sparse linear system, i.e. lots of coeficients are null. Non-zero coefficients appear only when the function chosen at one step is correlated with previously selected function(s). The system can be solved by the *conjugate gradient method.* This method is a general method for solving linear systems but it is especially attractive and efficient for large sparse systems [4].

In the context of DFD coding, it appears that the "in the loop" back-projection process only slightly improves the mean value of the PSNR along the sequence. Typically, improvements in the range 0.1 - 0.2 dB occur, e.g. for hall-monitor at 10 kbits/s, PSNR-Y is improved of 0.12 dB. Actually, improvements only occur when lots of correlated atoms are chosen in the same picture area. In this case, improvement is significant. In other cases, back-projection is useless. Nevertheless, as stated before, the computational cost of the back-projection is low and proportional to the correlation between the atom that have been selected. So, complexity is increased only if significant improvement is possible, depending on the sequence to code.

3 Dictionary Choice: Towards a Subband Dictionary

From the reminder of Mallat's MP description (see section 1), it appears clearly that any complete subset of $L^2(r)$ may be used as a dictionary. The energy conservation will always ensure the convergence of the iterative process. However, this convergence will be increased if the waveforms of the dictionary match the expended signal structures. In a coding context, fast convergence means few significant coefficients to transmit. A way to improve convergence is to increase the size of the dictionary but, of course, it will increase the bit budget allocated to the definition of the waveforms that are actually used to represent the signal. So, more than a computational complexity constraint, a coding efficiency constraint will restrict the size of the dictionary. An optimal dictionary will thus contain a limited number of waveforms that match well the common prediction error structures. As prediction error is localized, the waveforms should have a good spatial localization.

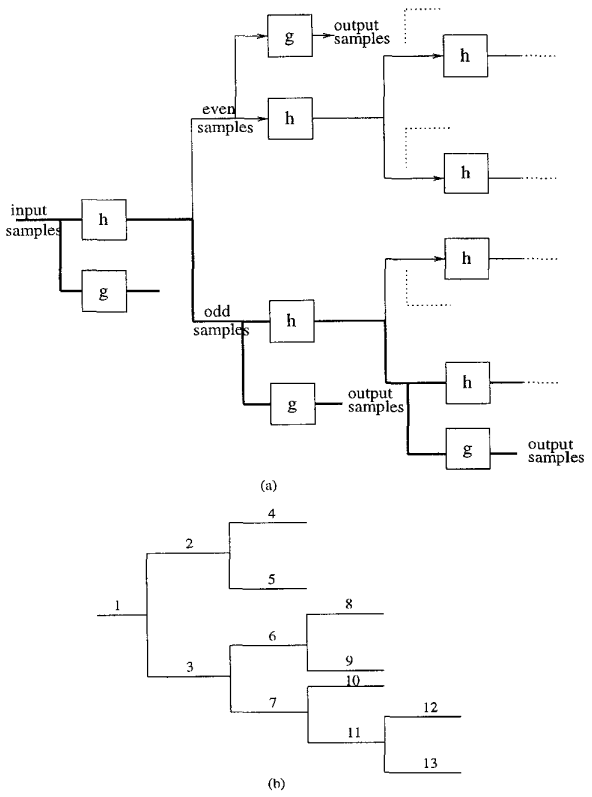


Figure 1: (a) Structure for the undecimated Discrete Wavelet Transform. Bold lines refer to the classical decimated wavelet transform structure. (b) Structure of the decimated Discrete Wavelet Packet Transform whose undecimated version defines the MP dictionary functions.

As a conclusion, it appears that there exist few constraints to fix the choice of the dictionary. A good choice could be to use a dictionary that reduce the complexity of the iterative process. The costliest part of the process is the computation of the inner products between dictionary functions and the signal to decompose. This inner product or correlation calculation is equivalent to a filtering procedure. MP should strongly benefit from the choice of a set of dictionary functions whose equivalent filter bank can be described through an efficient computational structure. It is worth noting that the large efforts carried out in order to provide efficient implementations of the wavelet transform [5] may be exploited to build computationally efficient dictionaries for MP. The algorithms that are discussed here, i.e. the "à trous" algorithm [6] and the undecimated version of the multiresolution wavelet representation algorithm [7], are filter bank structures. These structures perform an undecimated discrete wavelet transform [5] and can

	Sequences QCIF, 7.5 frames/s, 10kbits/s	
	Hall-Monitor	Mother-Daughter
Gabor Dictionary	PSNR-Y = 31.27 dB PSNR-U = 35.96 dB PSNR-V = 37.92 dB	PSNR-Y = 32.92 dB PSNR-U = 38.84 dB PSNR-V = 39.39 dB
Low cost subband dictionary	PSNR-Y = 31.32 dB PSNR-U = 35.94 dB PSNR-V = 37.95 dB	PSNR-Y = 32.87 dB PSNR-U = 38.65 dB PSNR-V = 39.21 dB

Figure 2: Comparison of the mean PSNR values obtained with both dictionaries.

be represented by the diagram of figure 1.a. h (g) is a low-pass (high-pass) filter.

In figure 2, a comparison is performed between the "classical" 400 2-D Gabor functions dictionary and a *subband dictionary*, i.e. a dictionary whose functions results from a tree structured filter bank. Actually, Haar functions have been used as h and g filters and the wavelet structure of figure 1.a has been generalized to the wavelet packet structure of figure 1.b. The impulse response of each node of this tree structure has been used as a 1-D dictionary function. The new dictionary contains 169 (13×13) functions. Its major interest versus the Gabor dictionary is its computational low complexity. During the DFD expansion process, each time an atom is searched for, inner products between the dictionary functions and the signal to expand have to be performed, the atom being defined by the maximum inner product value. With Gabor functions, using a 16×16 search window, 1733312 multiplications and about the same number of additions have to be performed each time an atom is searched for (see [2]). With the subband dictionary, this number is strongly reduced due to the tree structure of the filter bank.

Let S be the size of the search window. For the undecimated version of the transform, L being the length of the h and g filters (here, $L = 2$), computation of S samples at the i^{th} level of the tree requires the knowledge of $S + \sum_{l=j}^{i-1} 2^l(L-1)$ $j < i$ samples at the j^{th} level of the tree. This implies that, in order to compute S samples at the 4^{th} level of the tree, $S + 15$ samples have to be available at the 0^{th} level, i.e. on the non filtered data. At each node of the tree, when using Haar functions, one addition (or subtraction) and one multiplication by the $1/\sqrt{2}$ normalization factor have to be performed for each computed sample. So, the number of operations for filtering the data in the vertical direction is $V_{step} = (S+15).\sum_{n=2}^{13} 2.NS_n$ with $S+15$ = number of rows that have to be pre-computed before horizontal filtering, n = indices of the node of the filter in the tree

structure and NS_n = number of samples computed at node n. Developing this formula, we have $V_{step} = (S+15).[2.(S+14).2+2.(S+12).4+2.(S+8).4+2.S.2]$. For each of the 13 sets of $S.(S+15)$ vertically precomputed data, horizontal filtering is performed. The number of operations is $H_{step} = 13.S.(\sum_{n=2}^{1} 32.NS_n)$. With $S = 16$, $T_{step} = V_{step} + H_{step}$ results in 143400 operations, i.e. 71700 multiplications and the same number of additions. Nevertheless, for MP expansion purposes, the number of operations can still be reduced. Indeed, with a subband approach using Haar filters, the normalization factor only depends on the depth of the node in the tree structure. Actually, h and v being the depth of the horizontal and vertical component of the 2-D dictionary function, the normalization factor is $C = (1/\sqrt{2})^h.(1/\sqrt{2})^v = (1/\sqrt{2})^{h+v}$. An efficient implementation can benefit from this observation. In a first step, coefficients are computed only using additions or subtractions (as if the filters where $(1,1)$ and $(1,-1)$ instead of $(1/\sqrt{2}, 1/\sqrt{2})$ and $(1/\sqrt{2}, -1/\sqrt{2})$). After this, coefficients of nodes with the same $v+h$ are grouped. For each of the 9 groups, the maximum coefficient is extracted. Only these 9 maximum coefficients are normalized to extract the maximum one. As a conclusion, the extraction of an atom only requires 71700 additions (or subtractions) and 9 multiplications.

The 71700 additions and 9 multiplications of the subband dictionary have to be compared to the 1733312 multiplications and 1733312 additions of the Gabor dictionary. Assuming that additions are as costly as multiplications, this means that the subband dictionary permits a complexity reduction by a factor of nearly 50! This reduction concerns the most expensive part of the MP texture coding. So, it is significant.

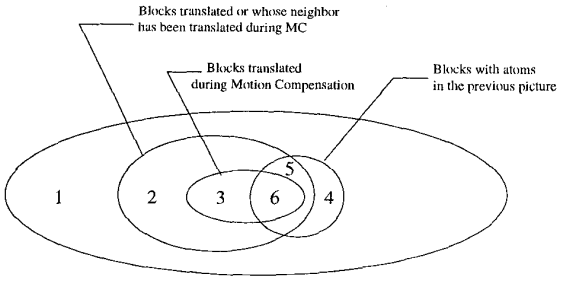
Note: to produce results of figure 2, the MPEG-4 Verification Model Software has been used as a basic platform and the rate control has been synchronized to the one of the MPEG-4 VM4 regulation model according to the method described by Neff and Zakhor [2]. The Atom parameters have been entropy coded as described in the next section.

4 Entropy Coding

4.1 Positions

Actually, the information defining the atoms distribution map is divided in three parts. The map being split in 8×8 blocks, the first part distinguishes the blocks containing atom(s) from others. The second part tells the number of atoms in each blocks. The third part specifies the atoms positions in each non empty block.

Some correlation exists between the amplitude of the motion field used to predict the current frame and the position of the atoms used to expend the prediction error. Besides, atoms will preferably be located in areas containing atoms in the previous frame. The splitting of the picture into 8×8 blocks is a simple way to take into account this *a priori* knowledge. Indeed, atoms are localized around moving areas and in



(a)

		Hall-Monitor			
		10 kbits/s		20 kbits/s	
Number of atoms		2148		7978	
Total bit budget for atom coding		40868		138033	
Position	Blocks with atoms	4750 (7915)*		8448 (14171)*	
	Number of atoms for each block with atoms	2766	Σ= 17748	8151	Σ= 51558
	Position of atoms in block	10232		34959	
	Berkeley	21287 (+16.6%)		62325 (+17.3%)	
YUV type		657		5540	
Function indices		14503		55875	
Amplitude		7960		25060	

(b) * in braces, results without categories distinction

Figure 3: (a) Definition of the six block categories, (b) Bit budget allocated to the coding of the atom parameters for sequences "Hall-Monitor".

areas with atoms in the previous picture. For each block, some information is available: does the corresponding block in the previous picture contain atoms? Has the block been translated during motion compensation? If not, has one of its neighbours been translated? Using this information, blocks in the current picture are classified in 6 categories summarized in figure 3.a. Each category is characterized by the probability p_A that any of its blocks contains atoms. For each picture and for each category, a binary sequence (one digit by block of the category) telling whether the corresponding block contains atom(s) or not will be encoded through an adaptive truncated run length (ATRL) code [8].

For blocks containing atoms, the number of atoms is specified. If $N(i)$ is the number of atoms in the ith block with atoms, the sequence of positive numbers $N(i) - 1$ is entropy encoded through an Universal Variable Length Coder [9]. This coder is based on the mapping of the numbers into binary memoryless sources encoded by ATRL.

At this step, for each block containing atoms, the exact positions of atoms in the block has to be defined. Again, this will be done through run length encoding of the binary sequence resulting from an original scanning of the pixels of the blocks containing atom(s). In these blocks, atoms are more probably located on the border of the block or on the edge of the predicted signal. So, inside a block, pixel should be sorted in decreasing order of their probability to support an atom. For the preliminary results presented here, pixels have been classified in decreasing order of their local gradient value (= sum of the absolute luminance difference between the pixel and each of its neighbors). In order to favour pixels along the border of the block, their gradient value is doubled. The sorted pixels are then scanned to build a binary sequence that will be run length encoded. This scanning goes over the blocks in a cyclic way, examining a single pixel of each block at a time. For each scanned pixel, a number of 1-symbol equal to the number of atoms the pixel supports is added to the binary sequence. Then a 0-symbol is added to the binary sequence to tell to go on the scanning. Of course, as the total number of atoms is known for each block, when this number is achieved for a block, remaining pixels of the block do not need to be scanned any more. They can be skipped.

4.2 YUV space, Function Indices and Magnitude

The type (Y,U,V) of each atom is encoded using ATRL. Fixed Huffman tables are used to code the functions indices. In order to fit the statistics of a large range of picture, a single huffman table is not sufficient. Indeed, efficient motion compensation results in high frequency and sharply located prediction errors. Functions with a short spatial extent are thus more probable. On the other hand, at lower qualities, when the prediction error is spread on large areas, functions with a larger spatial extent best match the error signal. As a result, we decided to use a multi-huffman coding strategy. Moreover, an attempt has been made to take into account some *a priori* knowledge deduced from the predicted picture about the shape of the prediction error in the surrounding of an atom location. Indeed, knowing the predicted picture and the atom location, some function shapes become more probable than others, e.g. along a vertical edge, vertical atoms are more probable than horizontal ones. This brings 4-5 % coding efficiency improvement. The coefficients magnitudes are coded in reference to the lower magnitude in the picture (transmitted on six bits), using **fixed** Huffman codes. Of course, one bit is used to code the sign of the coefficient amplitude.

4.3 Entropy Coding Results

Table of figure 3.b presents the coding cost of the atom distribution map. A comparison is performed between the method proposed by Neff and Zakhor [2] and the one proposed in section 4.1. Significant improvements occur (more than 15 %). It is worth noting that a more effective method has been developped in the MPEG-4 group for the "combine shape, motion and texture" coding mode (see MPEG-4 VM description). It also exploits the correlation that exist between picture motion and atom location. It produces results that are close to the one presented here. Moreover, it is block-based and more flexible. Authors recommend thus the reader to inform himself about this method which is also more suitable for object-based coding.

5 Conclusion

This paper has achieved one major result: the definition of a new subband dictionary that is both effective and efficient. Two other lessons can be drawn from this work: the atom distribution map coding efficiency can be increased by taking into account some a priori knowledge about the picture content, the orthogonalization of the signal expansion projection is low cost but only slightly improves coding quality.

References

[1] S. Mallat and Z. Zhang. Matching Pursuits with Time-Frequency Dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, December 1993.

[2] R. Neff and A. Zakhor. Very Low Bit Rate Video Coding Based on Matching Pursuits. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1):158–171, February 1997.

[3] M. Banham and J. Brailean. A Selective Update Approach to Matching Pursuits Video Coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1):119–129, February 1997.

[4] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. Solution of Linear Algebraic Equations. In *Numerical Recipies in C: The Art of Scientific Computing*, chapter 2, pages 78–90. Cambridge University Press, The Pitt Building, Trumpington Street, Cambridge CB2 1 RP, 1992 (Second Edition). ISBN 0-521-43108-5.

[5] M. Shensa. The Discrete Wavelet Transform: Wedding the 'à trous' and Mallat Algorithms. *IEEE Transactions on Signal Processing*, 40(10):2464–2482, October 1992.

[6] M. Holschneider, R. Kronland-Martinet, J. Morlet, and Ph. Tchamitchian. A Real-Time Algorithm for Signal Analysis with the Help of the Wavelet Transform. In *Wavelets: Time-Frequency methods and phase Space*, pages 286–297. Springer, IPTI, Berlin, 1989.

[7] S. Mallat. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[8] H. Tanaka and A. Leon-Garcia. Efficient Run Length Encoding. *IEEE Transactions on Information Theory*, 28(6):880–890, November 1982.

[9] B. Macq. A universal entropy coder for transform or hybrid coding. *Picture Coding Symposium*, pages 12.1.1–12.1.2, 1990.