A FAST TWO-STAGE ALGORITHM FOR REALIZING MATCHING PURSUIT

Kin-Pong Cheung and Yuk-Hee Chan

Center for Multimedia Processing
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong

ABSTRACT

Matching pursuit is proved to be one of the most efficient techniques in various signal coding techniques. However, its computation effort is so tremendous that it may not be affordable in some real-time applications. A two-stage filtering structure for realizing matching pursuit has recently proposed. This paper presents an efficient algorithm for the implementation of the structure. The issues of how to determine the elementary dictionary and approximate a given initial dictionary are also addressed in this paper. Simulation results show that the proposed algorithm can achieve a significant improvement in the complexity-distortion performance.

1. INTRODUCTION

The theory of matching pursuit (MP) was originally proposed by Mallat and Zhang [1] and has been successfully applied to and found its potential applications in video coding, image coding, speech coding and related areas [2-5]. Matching pursuit iteratively expands a signal onto an overcomplete set of basis functions and hence its realization complexity is tremendously high. A number of solutions have been proposed to reduce its computation effort. Most of them focus on looking for a simplified dictionary to reduce the effort for searching an appropriate basis vector while maintaining a reasonable coding quality in terms of rate distortion performance [6-8]. In other words, these fast algorithms are for dedicated dictionaries only. Recently, Redmill, Bull and Czerepinski proposed a 2-stage filtering structure for implementing matching pursuit [9]. This algorithm decomposes a basis vector of the dictionary into a weighted summation of basis vectors from an elementary dictionary of much smaller size. By doing so, one can compute the target inner products as weighted summations of the elementary inner products. Note this is basically a general fast algorithm for realizing matching pursuit as it is not restricted for dedicated applications and can be used with any given dictionary.

Based on this structure, Neff and Zakhor introduced a design methodology which incorporates both coding efficiency and complexity in a systematic way [10]. A mechanism was introduced to generate an approximation of the dictionary. In their approach, matching pursuit is used to decompose the basis vectors of the dictionary into weighted summations of basis vectors selected from an elementary dictionary. Though this 2-stage filtering structure was originally proposed for residual video coding, it can also be used in general applications.

In this paper, we will provide a systematic analysis of a general 2-stage filtering structure for realizing matching pursuit. Then we will suggest a systematic approach to construct the 2-stage filtering structure directly without using matching pursuit. We will also suggest a method to determine an elementary dictionary based on the given dictionary. Note the aim of the proposed method is not for particular applications such as residual video coding. It is a general technique for reducing the computational effort required by matching pursuit and can work with other proposed techniques such as dictionary simplification to reduce the complexity further.

2. BASIC MATCHING PURSUIT

The basic MP algorithm [1] decomposes a given column vector $\vec{x}=(x_1,x_2,....,x_K)^T$ into a weighted summation of vectors from a dictionary of S K-dimensional unit basis vectors, say $\Omega=\{\vec{d}_l \mid l=1,2..S\}$, in an iterative manner. This dictionary is referred to as initial dictionary in [10] and we follow the convention in this paper. The decomposition is carried out by successive approximation. Let $\vec{r}_j$ be the residue, which is defined to be the difference between $\vec{x}$ and the approximated output, after the j^{th} iteration. At the $(j+1)^{th}$ iteration, the basis vector $\vec{d}^*_{j+1}\in\Omega$ that best matches $\vec{r}_j$ in a sense that their inner product is maximum is selected. In formulation, we have

$$\vec{d}^*_{j+1}=\max_{\vec{d}_l\in\Omega}\left|\vec{d}_l^{\,T}\vec{r}_j\right| \tag{1}$$

where $\max_{\vec{d}_l\in\Omega} C$ denotes the vector $\vec{d}_l$ which belongs to Ω and maximizes cost function C. The residue of this stage is then given by

$$\vec{r}_{j+1}=\vec{r}_j-\left(\left(\vec{d}^*_{j+1}\right)^T\vec{r}_j\right)\vec{d}^*_{j+1} \tag{2}$$

This process repeats until $\|\vec{r}_j\|$ is smaller than a predefined threshold or some other criteria are met.

Suppose now we have an elementary dictionary $\Omega'=\{\vec{t}_i \mid i=1,2..S'\}$ and we approximate all $\vec{d}_l$'s in Ω with $\vec{d}'_l$'s, where

$$\vec{d}'_l=\sum_{i=1}^{S'}\beta_{il}\vec{t}_i\ . \tag{3}$$

Then, we have $\vec{d}_l'^{\,T}\vec{r}_j=\sum_{i=1}^{S'}\beta_{il}(\vec{t}_i^{\,T}\vec{r}_j)$. The realization of matching pursuit then becomes a 2-stage process as shown in Fig. 1. In the first stage, $\alpha_{ij}=\vec{t}_i^{\,T}\vec{r}_j$ is computed for all $\vec{t}_i\in\Omega'$. In the second stage, $\gamma_{lj}=\sum_{i=1}^{S'}\beta_{il}\alpha_{ij}$ are computed. The basis vector $\vec{d}_l$ whose associated $|\gamma_{lj}|$ is maximum is selected. This structure was proposed in [9].

0-7803-6725-1/01/$10.00 ©2001 IEEE

There are two basic issues about the structure. The first one is what the optimal Ω' should be, which was not addressed in either [9] or [10]. The second one is how to approximate $\vec{d}_l$'s in Ω with $\vec{t}_i$'s in Ω'. Neff and Zakhor suggested using matching pursuits [10]. In the following section, we will suggest a simple method to determine the optimal Ω' and do the approximation directly and effectively.

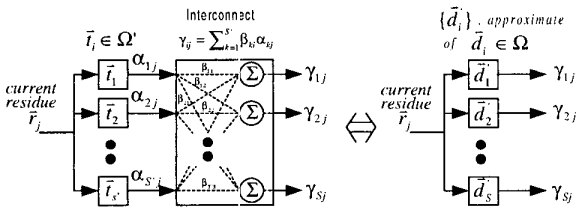

Fig. 1 Connection between a basis vector and the residue

3. PROPOSED MP ALGORITHM

By having a close look at the structure shown in Fig. 1, one can see that the first stage of the structure is actually extracting features of the input residue with different operators for the second stage to process. Obviously, the $\vec{t}_i$'s in Ω' should be uncorrelated with each other. Otherwise, there must be some redundancy in the features described by $\vec{t}_m{}^T \vec{r}_i$ and $\vec{t}_n{}^T \vec{r}_i$, where $m \neq n$. This implies inefficiency in the first stage of the structure. Hence, $\vec{t}_i$ should be selected in a way that $\vec{t}_m{}^T \vec{t}_n = 0$ if $m \neq n$. Without losing generality, we can assume that $\vec{t}_i$'s are unit vectors. In such a case, we have $T^T T = I$, where I is the identity matrix and $T = (\vec{t}_1, \vec{t}_2, ..\vec{t}_{S'})$.

In order to make the best approximation of $\vec{d}_l$ with $\vec{t}_k$'s in Ω', one has to minimize the following objective function

$$J = \sum_{\vec{d}_l \in \Omega} \| \vec{d}_l - \vec{d}_l' \|^2 = \sum_{\vec{d}_l \in \Omega} \| \vec{d}_l - T\vec{b}_l \|^2 \qquad (4)$$

where $\vec{b}_l = (\beta_{1l}, \beta_{2l} ... \beta_{S'l})^T$. By solving $\frac{\partial J}{\partial \vec{b}_l} = \vec{0}$, we have

$$\vec{b}_l = T^T \vec{d}_l \qquad (5)$$

By substituting (5) into (4), we have

$$J = \sum_{\vec{d}_l \in \Omega} \| \vec{d}_l - TT^T \vec{d}_l \|^2$$
$$= \sum_{\vec{d}_l \in \Omega} (\vec{d}_l{}^T \vec{d}_l) - \sum_{\vec{d}_l \in \Omega} (\vec{d}_l{}^T TT^T \vec{d}_l) \qquad (6)$$

Since both terms are larger than zero, minimizing J is equivalent to maximizing $J_1 = \sum_{\vec{d}_l \in \Omega} (\vec{d}_l{}^T TT^T \vec{d}_l)$, which is actually the total energy of $T^T \vec{d}_l$'s. By reformulating J_1, we have

$$J_1 = \sum_{\vec{t}_i \in \Omega'} \left(\vec{t}_i{}^T \left(\sum_{\vec{d}_l \in \Omega} (\vec{d}_l \vec{d}_l{}^T) \right) \vec{t}_i \right) = \sum_{\vec{t}_i \in \Omega'} (\vec{t}_i{}^T (E\lambda E') \vec{t}_i) \qquad (7)$$

where λ is a diagonal matrix of the sorted eigenvalues of $A = \sum_{\vec{d}_l \in \Omega} (\vec{d}_l \vec{d}_l{}^T)$, the autocorrelation matrix of $\vec{d}_l \in \Omega$, and E is a matrix whose columns are the corresponding eigenvectors so that $AE = E\lambda$. Obviously, to maximize J_1 with fixed S', $\vec{t}_i$'s should be the eigenvectors of A and their corresponding eigenvalues must be of the largest magnitude among the eigenvalues of A. In other words, the elementary dictionary Ω' can be determined by selecting the most significant S' principal components of $\vec{d}_l$'s in Ω. When S' is equal to or larger than the rank of A, $\vec{d}_l \in \Omega$ can be perfectly reconstructed with $\vec{t}_i$'s. This implies that the maximum size of Ω' is the rank of A, which is much smaller than the size of the initial dictionary. Note this reduction of complexity does not introduce any distortion to the encoded signal.

With the elementary dictionary obtained with our proposed method, a straightforward method for approximating $\vec{d}_l$'s is proposed as follows. After getting $\vec{b}_l = (\beta_{1l}, \beta_{2l} ... \beta_{S'l})^T$ with eqn. (5), the β_{kl}'s of smallest non-zero magnitude are masked by setting them to zero one by one until the sum of squares of those left behind is not larger than $\kappa \| \vec{b}_l \|^2$, where κ is the quality index specified and $1 \geq \kappa \geq 0$. This is equivalent to breaking the corresponding physical connections in the second stage of the filtering structure. Note, if $\vec{d}_l$'s are highly correlated, there will be only a few of significant β_{kl}'s and the reduction in the complexity will be significant as the method using matching pursuit does.

In summary, our proposed 2-stage matching pursuit algorithm can be implemented as follows.

Initialization

Step 1: Determine the principal components of the basis vectors in dictionary Ω and form the transform kernel $T_c = (\vec{t}_1, \vec{t}_2, ... \vec{t}_K)$, where $\vec{t}_i$ is the i[th] most significant principal component.

Step 2: Select $\{\vec{t}_i \mid i = 1,2..S'\}$ to form the elementary dictionary Ω'.

Step 3: Precompute $\vec{b}_l = T_c{}^T \vec{d}_l$ for all $\vec{d}_l \in \Omega$.

Step 4: Mask all insignificant principal components in $\vec{b}_l$ based on the selection result of Step 2.

Step 5: Further mask all insignificant weights in $\vec{b}_l$ according to the given quality index κ to get $\vec{b}'_l$.

Realization

Stage 1: Compute $\vec{a}_0 = T_c{}^T \vec{x}$.

Stage 2: Do the following steps repeatedly until the terminating criterion is satisfied:

1. Compute $\gamma_{lj} = (\vec{b}'_l)^T \vec{a}_j$ (8)

2. Select the $\vec{d}_l$ whose $|\gamma_{lj}|$ is maximum as $\vec{d}^*$
3. Update $\vec{a}_j$ with

$$\vec{a}_{j+1} = \vec{a}_j - \left(\left(\vec{b}^*\right)^T \vec{a}_j \right) \vec{b}^* \qquad (9)$$

where $\vec{b}^* = T_c^T \vec{d}^*$ is precomputed

Note eqn. (8) can be rewritten as $\gamma_{lj} = \left(\vec{b}'_l\right)^T \vec{a}_j$ $= \left(\vec{b}'_l\right)^T \vec{a}_{j-1} - \left(\left(\vec{b}^*\right)^T \vec{a}_{j-1} \right) \left(\left(\vec{b}'_l\right)^T \vec{b}^* \right)$. Hence, γ_{lj} can be computed with only 1 multiplication and 1 addition for each l if $j>0$ as $\left(\vec{b}'_l\right)^T \vec{b}^*$ can be precomputed, and, $\left(\vec{b}'_l\right)^T \vec{a}_{j-1}$ and $\left(\vec{b}^*\right)^T \vec{a}_{j-1}$ were computed in previous iteration (j-1).

Besides, the approximation of $\vec{d}_l$ made in our approach is for reducing the complexity of computing (8) only. The same initial dictionary is actually used for matching pursuit in the encoder. This is obviously an advantage as compared with approaches which use modified dictionaries [6-8] or approximated dictionaries [10] as the dictionaries used in their decoders have to be modified accordingly to match the dictionaries used in their encoders. There is no such a constraint in our proposed approach and hence it can be applied to any existing MP codecs to reduce the complexity of their encoders without affecting their corresponding decoders.

4. SIMULATION RESULT

A simulation was performed to evaluate the efficiency of the proposed MP algorithm. The basis vectors used for constructing the dictionary Ω are masked version of 400 predefined Gabor functions [2]. The basis vectors are of size 8x8 each. Fig. 2 shows the packed basis vectors.

The proposed MP algorithm was tested with different standard images including *Lenna, House, Girl, Germany* and *Baboon*. The elementary dictionary Ω' was first determined with the proposed method. The dictionary Ω was then approximated in accordance with different quality indices κ by using Neff's[10] or our approaches. Corresponding two-stage filtering structures were then formed.

In the simulation, images were divided into a number of 8x8 blocks and each block was then approximated with a fixed number of basis vectors by using matching pursuits with a particular two-stage filtering structure. Fig. 3 shows the average number of multiplications required to achieve a particular average residual energy level when our approximation method is used to implement the 2-stage filtering structure. Note that, similar to most matching pursuit algorithms, the number of multiplications required is more or less the same as that of additions in both concerned approaches. Hence, Fig. 3 actually reflects the complexity-distortion performance of the proposed approach. Fig. 3a shows the case where the elementary dictionary contains all principal components of $\vec{d}_l$'s while Fig. 3b shows the case where only the most significant half of them are contained. One can see that there is no significant difference in the performance achieved when $\kappa < 1$. This is because, after the transformation, energy is compacted and most of the principal components are insignificant. The curves marked as 'F' and 'F+' in the Figure show, respectively, the performance of the direct implementation of matching pursuits and that of the implementation with the fast updating formula referred to as eqn. (33) in [1]. They are plotted for reference.

Fig. 4 shows the case when Neff's method is used. The elementary dictionary Ω' used in the simulation was determined with our proposed method. By comparing Fig. 3 and Fig. 4, it can be found that the performance of our structure is better than Neff's. The complexity is significantly reduced for achieving a particular residual energy level. Note that the effect of the quality index κ in Neff's approach is different from that in ours and hence it is meaningless to compare a curve in Fig. 3 with a curve obtained with identical value of κ in Fig. 4.

Note the proposed approach is not mutually exclusive with some other useful techniques and can work with them to reduce the complexity further. For example, one can use the technique proposed in [11] to convert a large numbers of multiplications into additions without introducing any additional distortion.

5. CONCLUSION

In this paper, a systematic analysis of the 2-stage filtering structure proposed in [9] is provided. A systematic approach is then proposed to construct the 2-stage filtering structure. Unlike the approach in [10], the proposed approach constructs the structure directly without using matching pursuit. A method is also suggested to select an optimal elementary dictionary based on the initial dictionary. Simulation results show that the complexity-distortion performance can be much improved as compared with [10].

6. ACKNOWLEDGEMENT

The work described in this paper is substantially supported by RGC Grant G-V763 and the Center for Multimedia Signal Processing, HKPolyU.

7. REFERENCES

[1] S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries," *IEEE Trans. Signal Processing*, Vol. 41, pp.3397-3415, Dec 1993.

[2] 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.

[3] Osama K. Al-Shaykh, E. Miloslavsky, T. Nomura, R. Naff and A. Zakhor, "Video compression using matching pursuit," *IEEE Trans. Circuits Syst. Video Technol.*, Vol.9, pp.123-143, Feb 1999.

[4] Y.H. Chan, "An Efficient Weight Optimization Algorithm for Image Representation using Nonorthogonal Basis Images," *IEEE Signal Processing Letters*, Vol. 5, No. 8, pp. 193-195, Aug 1998.

[5] R. Gribonval, E. Bacry, S. Mallat, P. Depalle and X. Rodet, "Analysis of sound signals with high resolution matching pursuit," Proceedings of the IEEE-SP Intl. Symposium on Time-Frequency and Time-Scale Analysis, pp.125 -128, 1996.

[6] P. Czerepinski, C. Davies, N. Canagarajah and D. Bull, "Matching pursuits video coding: dictionaries and fast implementation," *IEEE Trans. Circuits Syst. Video Technol.*, Vol.10, No.7, pp.1103-1115, Oct 2000.

[7] Q. Liu, Q. Wang and L. Wu, "Dictionary with tree structure for matching pursuit video coding," *Electronics Letters*, Vol.36, No.15, pp.1266-1268, 20 July 2000.

[8] C.D. Vleeschouwer and B. Macq, "Subband dictionaries for low-cost matching pursuits of video residues", *IEEE Trans. Circuits Syst. Video Technol.*, Vol.9, No.7, pp.984-993, Oct 1999.

[9] D.W. Redmill, D.R. Bull and P. Czerepinski, "Video coding using a fast non-separable matching pursuits algorithm," Proceedings, IEEE ICIP'98, Vol.1, pp.769 -773, 1998.

[10] R. Neff and A. Zakhor, "Dictionary approximation for matching pursuit video coding," Proceedings, IEEE ICIP'00, Vol.II, pp.828-831, 2000.

[11] K.P. Cheung and Y.H. Chan, "An efficient algorithm for realizing matching pursuits and its applications in MPEG4 coding system," Proceedings, IEEE ICIP'00, Vancouver, Canada, Vol. II, pp.863-866, 2000.

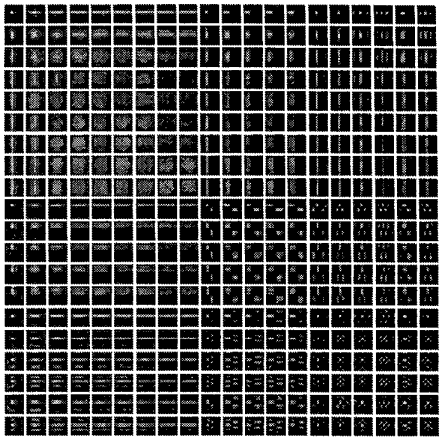

Fig. 2 The 2D windowed Gabor Dictionary

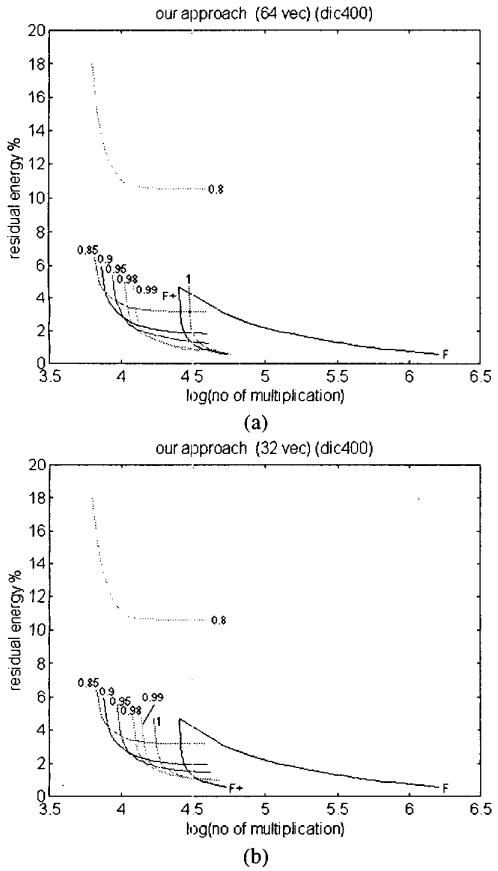

Fig. 3 The complexity-distortion performance of the proposed approach at different quality index κ when size of Ω = 400. (a) Size of Ω' = 64; (b) Size of Ω' = 32.

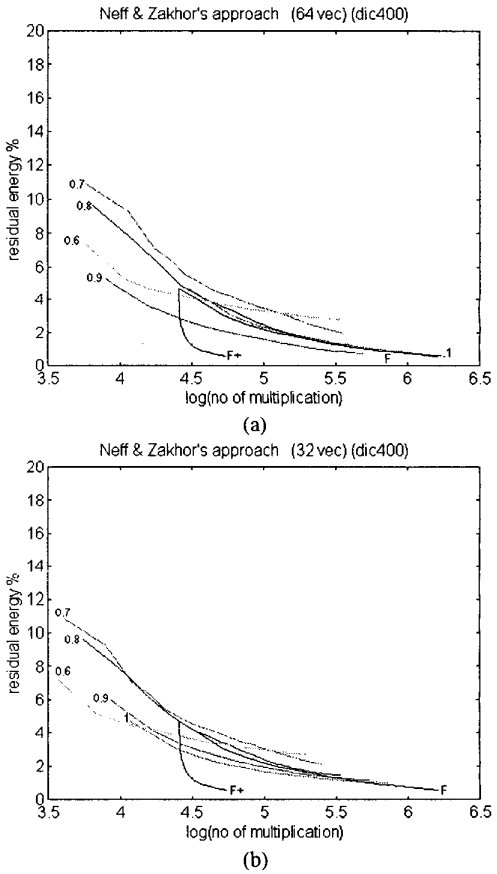

Fig. 4 The complexity-distortion performance of Neff & Zakhor's approach at different quality index κ when size of Ω = 400. (a) Size of Ω' = 64; (b) Size of Ω' = 32.