AN EFFICIENT ALGORITHM FOR REALIZING MATCHING PURSUITS AND ITS APPLICATIONS IN MPEG4 CODING SYSTEM

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 Pursuits is proved to be one of the most efficient techniques in various video coding techniques. However, its computation effort is so tremendous that it may not be affordable in some real-time applications. This paper presents an efficient algorithm for the implementation of Matching Pursuits. Simulation results show that the computation effort can be successfully reduced with the proposed method.

1. INTRODUCTION

Matching Pursuits (MP) [1], an iterative approach to expand a signal onto an overcomplete set of basis functions, has been successfully applied in MPEG-4 video coder to code video sequences [2-3].

Although MP is shown to be able to achieve better PSNR and visual quality while maintaining similar decoding complexity as compared with MPEG-4 DCT codec [4], the high complexity of its encoder makes it less appealing for real-time applications.

In this paper, we provide a new concept that makes use of the partial searching technique [5] to realize MP. The aim of the proposed method is to boost the coding efficiency by reducing the computational effort as required by the original MP [1].

2. BASIC MATCHING PURSUITS

Basic MP algorithm [1] decomposes a given vector $\vec{x}=(x_1,x_2,....,x_k)$ into a weighted summation of vectors from a dictionary of N k-dimensional unit basis vectors, say $\Omega=\{\vec{v}_i \mid i=1,2..N\}$, in an iterative manner. 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 jth iteration. At the (j+1)th iteration, the basis vector $\vec{b}_{j+1}\in\Omega$ that best matches $\vec{r}_j$ in a sense that their inner product is maximum is selected. In formulation, we have

$$\vec{b}_{j+1}=\max_{v_l\in\Omega}\left|<\vec{v}_l,\vec{r}_j>\right| \quad (1)$$

where $<\vec{a},\vec{b}>$ denotes the inner product of vectors $\vec{a}$ and $\vec{b}$, and, $\max_{\vec{v}_l\in\Omega} C$ denotes the vector $\vec{v}_l$ which belongs to Ω and maximizes cost function C. The residue of this stage is then given by

$$\vec{r}_{j+1}=\vec{r}_j-\alpha_{j+1}\,\vec{b}_{j+1} \quad (2)$$

where

$$\alpha_{j+1}=<\vec{b}_{j+1},\vec{r}_j> \quad (3)$$

Note all $\vec{v}_i$'s in $\Omega\setminus\{\vec{b}_j\}$ should be tested to find the best match for $\vec{r}_j$. This process repeats until $\|\vec{r}_j\|$ is smaller than a pre-defined threshold or some other criteria are met. Tremendous computation effort is required for such an iteration process. Sometimes this makes MP not practical for real-time applications.

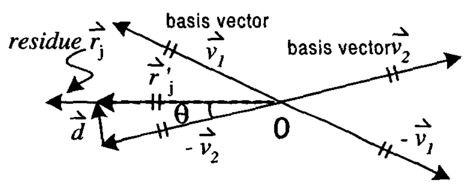

Figure 1. Connection between a basis vector and the residue

3. PROPOSED MP ALGORITHM

From eqn. (1), one can see that we are actually looking for a vector in Ω the projection of which or its negative on $\vec{r}_j$ is maximum. As shown in Figure 1, this is equivalent to finding a vector $\vec{u}_l$ which belongs to $\Omega'\equiv\{-\vec{u}\mid\vec{u}\in\Omega\}\cup\Omega$ and minimizes the following cost function

$$J=\|\vec{d}_{j,l}\|^2\equiv\|\vec{r}'_j-\vec{u}_l\|^2=2(1-<\vec{r}'_j,\vec{u}_l>) \quad (4)$$

where $\vec{r}'_j=\vec{r}_j/\|\vec{r}_j\|$.

Matching Pursuits can then be reformulated as another problem as, at the (j+1)th iteration, finding a unit vector $\vec{b}_{j+1}$ in Ω' which satisfies

$$\vec{b}_{j+1}=\min_{\vec{u}_l\in\Omega'}\|\vec{d}_{j,l}\|^2=\min_{\vec{u}_l\in\Omega'}\sum_{m=1}^{k}(r'_{j,m}-u_{l,m})^2 \quad (5)$$

where $r'_{j,m}$'s and $u_{l,m}$'s are, respectively, the mth elements of $\vec{r}'_j$ and $\vec{u}_l$, and get the residue

0-7803-6297-7/00/$10.00 © 2000 IEEE

$$\vec{r}_{j+1} = \vec{r}_j - \langle \vec{b}_{j+1}, \vec{r}_j \rangle \vec{b}_{j+1}$$
$$= \vec{r}_j - (1 - J_{\min}/2) \| \vec{r}_j \| \vec{b}_{j+1} \quad (6)$$

until the termination criterion is satisified. Here, $J_{\min} = \| \vec{r}'_j - \vec{b}_{j+1} \|^2$ and $\min_{\vec{u}_l \in \Omega'} C$ denotes the vector $\vec{u}_l$ which belongs to Ω' and minimizes cost function C.

By doing so, one can effectively convert a MP problem into a vector quantization (VQ) problem and make use of those well-developed fast VQ algorithms to reduce the computation effort for realizing MP. In particular, we can make use of the simple partial searching technique [5] to achieve the reduction of computation effort.

Figure 2 shows the basic structure of partial searching technique in pseudo code. It rejects the vector being tested as soon as the accumulated partial sum of product, say J_{acc}, is larger than or equal to a threshold $T_{\min}$ during evaluating $\| \vec{d}_{j,l} \|^2$. The encoder knows that this particular basis vector is not the right one before completing the evaluation of its cost. If the vector survives, its cost will become the new $T_{\min}$ for testing the vectors left.

```
initialize T_min;
for all u_l's in Ω'
    J_acc = 0;
    for (m = 1 to k)
        J_acc = J_acc + (r'_{j,m} - u_{l,m})^2;
        if (J_acc ≥ T_min) break;
    if (J_acc < T_min)
        T_min = J_acc;
        b_{j+1} = u_l;
```

Figure 2. Structure of partial searching technique

4. TRICKS FOR IMPROVING EFFICIENCY

It is not easy to conclude that the proposed approach can definitely reduce the computation effort in terms of number of operations. On one hand, the proposed approach successfully transforms the nature of the problem such that the partial searching technique can be exploited to reduce computation effort. On the other hand, that the size of the dictionary is doubled increases the number of vectors to be tested. As a matter of fact, if one cannot quit the inner loop shown in Figure 2 in a few steps, the computation effort required will be even more. Hence, tricks should be played to make the best use of the partial searching technique and reduce the number of vectors to be searched prior to the search or at the earliest stage. They are discussed as follows:

- Obviously, the earlier we test the right basis vector $\vec{b}_{j+1}$, the sooner we have $T_{\min} = J_{\min}$. More basis vectors can then be rejected in the early stage of the evaluation of their costs and hence more computation effort can be saved. Therefore, the search sequence should be well organized.
- In order to make $J_{acc} \geq T_{\min}$ happen sooner so as to quit sooner during evaluating the cost of an inappropriate $\vec{u}_l$, one should sort $(r'_{j,m} - u_{l,m})$'s for all $m \in \{1,2..k\}$ and accumulate their square values from the largest one to the smallest one.
- Note $\vec{u}_l$ and $-\vec{u}_l$ appear as a pair in Ω'. If $\| \vec{d}_{j,l} \|^2$ is the minimum at the moment, then we will have $\| \vec{r}'_j - \vec{u}_l \|^2 \leq \| \vec{r}'_j + \vec{u}_l \|^2$ and hence $-\vec{u}_l$ can be rejected immediately.
- Partial searching technique is not always positive. When the threshold $T_{\min}$ is very large, it is likely that we cannot reject an inappropriate $\vec{u}_l$ in a few of steps during the evaluation of its cost. Accordingly, the computational effort required will be very high. In such a case, it is better to evaluate the cost directly with $J = 2(1 - |\langle \vec{r}'_j, \vec{u}_l \rangle|)$, which will be able to test both $\vec{u}_l$ and $-\vec{u}_l$ simultaneously. A predefined threshold T_h is used to check if it is the case.
- One may also compare the signs of the elements of $\vec{u}_l$ with those of $\vec{r}'_j$. If the number of matches is smaller than a predefined threshold T_s, then we may assume $\vec{u}_l$ is unlikely to be the right vector and reject it directly. Note this assumption is not always true and hence sometimes it will reject the right vector though it seldom happens when an appropriate T_s is used. In this paper, the number of sign matches between vectors $\vec{v}_l$ and $\vec{r}'_j$ is defined as $S(\vec{v}_l, \vec{r}'_j) \equiv \sum_{m=1}^{k} \{sign(v_{l,m} \times r'_{j,m})\}$, where $sign(a) = \begin{cases} 1 & a > 0 \\ 0 \quad if & a = 0 \\ -1 & a < 0 \end{cases}$.
- From eqn. (4), we have $J_{\min} \leq 2$. Hence, any basis vector the accumulated partial sum of which is larger than 2 can be rejected immediately during evaluating its cost function. In fact, by assuming that the basis vectors are evenly distributed over the vector space, one may even make a more aggressive assumption that $J_{\min} < \frac{1}{N} \sum_{\vec{v}_l \in \Omega} \| \vec{v}_l - \vec{v}_{l^*} \|^2$, where $\vec{v}_{l^*} \in \Omega$ is the closest basis vector to $\vec{v}_l$. Note Ω is generally overcomplete and hence we have $\frac{1}{N} \sum_{\vec{v}_l \in \Omega} \| \vec{v}_l - \vec{v}_{l^*} \|^2 << 2$. This can be used as the initial value of $T_{\min}$ to reject basis vectors more effectively.

Figure 3 shows the flow of using partial searching technique to find $\bar{b}_{j+1}$ when a normalized residual input vector $\bar{r}'_j$ is given.

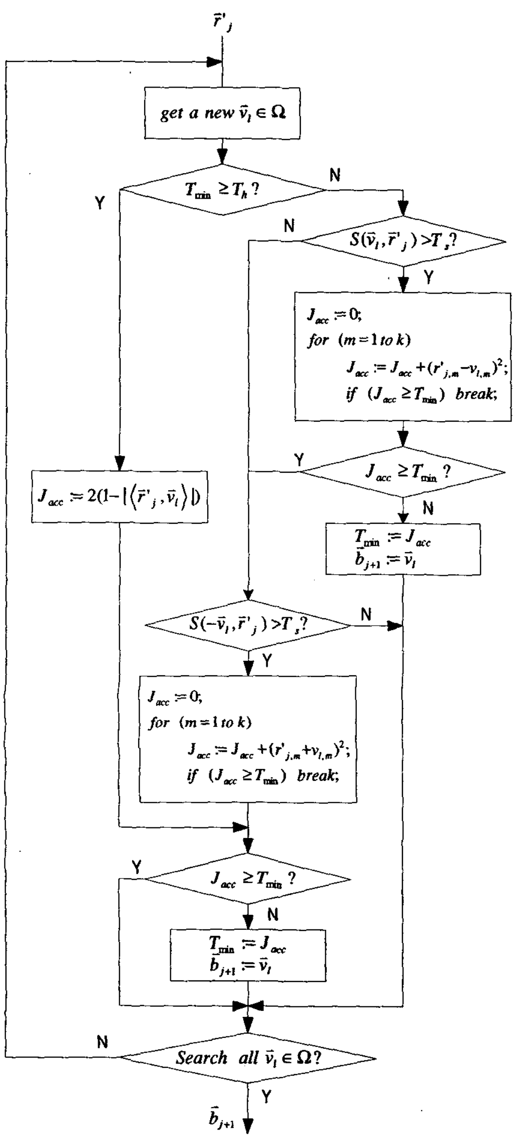

Figure 3. Flow of using partial searching technique to find a basis vector for representing a normalized residual vector

5. APPLICATIONS IN MPEG4 CODING SYSTEM

The video coding standard MPEG-4, which aims at providing many new features to cater for future multimedia applications, should have the ability to encode video objects of arbitrary shapes [6]. In conventional block-based image/video coding schemes [7,8], images are partitioned into blocks of pixels before encoding and image segments of arbitrary shapes cannot be coded efficiently. In order to achieve the goal, different coding techniques for encoding image segments of arbitrary shapes have been developed [9].

We applied the proposed MP algorithm to encode image segments of arbitrary shapes as in [3]. First, with the input of an image and its corresponding mask, image segments of arbitrary shapes are formed as follows. We divide a masked image into blocks of $N \times N$ pixels. Then block classification is performed. Three types of block are defined. They are foreground blocks, background blocks and boundary blocks. Foreground blocks are encoded with conventional block-based DCT coding technique. Background blocks contain useless information and hence are discarded. Boundary blocks, which consist of both foreground object and background, are coded with MP.

6. SIMULATION RESULT

A simulation is performed to evaluate the efficiency of the proposed MP algorithm. The coding scheme discussed in section 5 was used as a vehicle to evaluate the complexity of the algorithm. The basis vectors used for constructing a dictionary are masked version of 64 predefined orthogonal DCT vectors. The zigzag sequence employed in JPEG for encoding DCT coefficients was used as the searching sequence for $\bar{b}_j$'s in the realization of the proposed algorithm.

The proposed MP algorithm is tested with different standard sequences including *Kids*, *Weather Report*, *Akiyo*, *News* and *Bream*. In the simulation, thresholds T_h and T_{min} are, respectively, set and initialized to be 1.2. The first 150 frames of each testing sequence are encoded with either basic MP (BMP) [1] or the proposed algorithm(MMP). Figure 4 shows the average number of arithmetic operations required to achieve a particular residual energy level in all boundary blocks when the *Bream* sequence was encoded. One can see that the number of multiplications required for MMP is significantly reduced as compared with that of BMP at a cost of an increase of addition. Similar results were obtained with other testing sequences. Table 1 summaries the results of the case that the matching pursuit carried out in a boundary block stops when the energy of the residue is less than 5% of the energy of the foreground portion of the block. The figures shown in Table 1 are normalized with respect to the number of operations required for BMP. Unlike other modified MP algorithms such as [10], the output of the proposed algorithm is basically identical to that of the BMP as there is no difference in their basic structures.

For the evaluation of the coding performance of the aforementioned video coding scheme, DCT coding with mean padding technique (DCTWMP) [11] was simulated for comparison. Their PPSNR curves are shown in Figure 5. As expected, the quality of the coding scheme using MP is better than that of DCTWMP.

7. CONCLUSION

In this paper, an MP algorithm is proposed. This algorithm converts an MP problem into a VQ problem effectively. With such a conversion, one can make use of many well-developed fast VQ algorithms such as partial searching technique [5] to reduce the computation effort of the original MP. Simulation results show that the proposed algorithm can successfully convert a large number of multiplications required by the basic MP algorithm into additions. The same idea can be applied to other versions of MP to reduce their realization effort.

8. ACKNOWLEDGEMENT

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

9. REFERENCES

[1] S. Mallat and Z. Zhang, "Matching pursuits 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 pursuits," *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 pursuits," *IEEE Trans. Circuits Syst. Video Technol.*, Vol.9, pp.123-143, Feb 1999.

[4] Neff, R.; Nomura, T.; Zakhor, A., "Decoder complexity and performance comparison of matching pursuits and DCT-based MPEG-4 video codecs, " Proceedings, IEEE ICIP'98, Vol.1, pp.783 -787, 1998.

[5] Ngwa-Ndifor, J.; Ellis, T., "Predictive partial search algorithm for vector quantisation", *Electronics Letters*, Vol. 2719, pp. 1722 –1723, 12 Sep 1991.

[6] WG11 Document W1796, MPEG-4 Video Verification Model 8.0, Jul 1997.

[7] ITU Telecom. Standardization Sector of ITU, Video codec for audiovisual services at p×64 kbit/s," ITU-T Recommendation H.261, 1993.

[8] ITU Telecom. Standardization Sector of ITU, Video coding for low bitrate communication," Draft ITU-T Recommendation H.263, Version 2, 1998.

[9] T. Sikora and B.Makai, "Shape-adaptive DCT for generic coding of video," *IEEE trans. Circuits Syst. Video Technol.*, Vol. 5, pp. 59-62, Feb 1995.

[10] S.F. Cotter, R.Adler, R.D.Rao and K.Kreutz-Delgado, "Forward sequential algorithms for best basis selection," *IEE Proceedings-Vision, Image and Signal Processing*, Vol.146, Issue 5, pp.235 -244, Oct 1999.

[11] Joo-Hee Moon; Ji-Heon Kweon; Hae-Kwang Kim, "Boundary block-merging (BBM) technique for efficient texture coding of arbitrarily shaped object", *IEEE Trans. Circuits Syst. Video Technol.*, Vol.91, pp.35-43, Feb 1999.

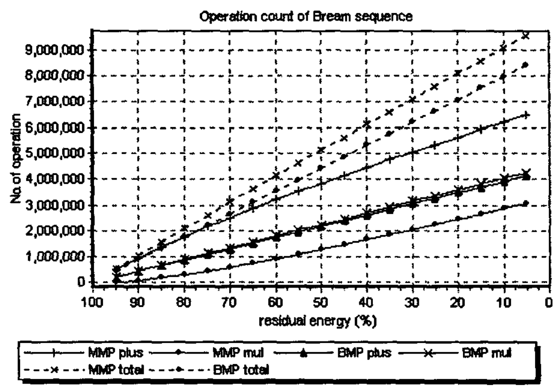

Figure 4. Average number of arithmetic operations required for Bream squence with $T_s = 0$

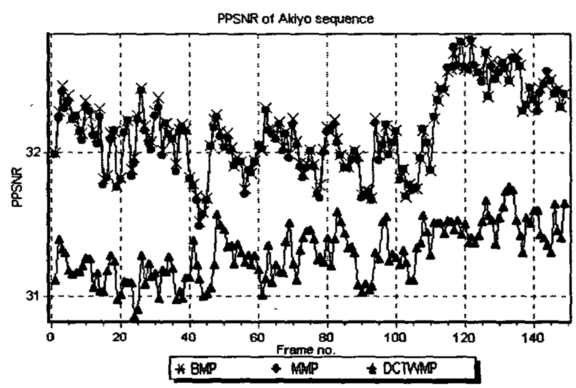

Figure 5. PPSNR of Akiyo sequence with $T_s = 0$

	BMP	MMP				
		$T_s = 0$	$T_s = -1$	$T_s = -2$	$T_s = -3$	$T_s = -4$
Akiyo :	BEF : 81		APB: 40557		MPB: 41818	
Add	1	+61%	+70%	+78%	+84%	+89%
Mul	1	-30%	-29%	-28%	-28%	-27%
Kids :	BEF : 146		APB: 42353		MPB: 43661	
Add	1	+63%	+73%	+81%	+88%	+93%
Mul	1	-32%	-31%	-30%	-29%	-29%
Bream :	BEF : 106		APB: 41159		MPB: 42420	
Add	1	+55%	+61%	+71%	+76%	+81%
Mul	1	-27%	-26%	-25%	-25%	-24%
News :	BEF : 112		APB: 40087		MPB: 41345	
Add	1	+59%	+67%	+71%	+80%	+86%
Mul	1	-28%	-26%	-27%	-26%	-25%

BEF : Average number of boundary blocks per frame
APB : Average number of additions per block required by BMP
MPB : Average number of multiplications per block required by BMP

Table 1. Average number of operations required for encoding the boundary blocks of a sequence.