THE FINE-GRAINED SCALABLE VIDEO CODING BASED ON MATCHING PURSUITS

Jian-Liang Lin[†*], *Wen-Liang Hwang*[†] *and Soo-Chang Pei*[*]

*Graduate Institute of Communication Engineering, National Taiwan University, Taiwan, R.O.C.
†Institute of Information Science, Academia Sinica, Nankang, Taipei, Taiwan, R.O.C.

ABSTRACT

Since channel capacity varies depending on the network traffic and the capacity of each reception, fine granularity scalability (FGS) of video coding has emerged over the past years as an important area in multimedia streaming applications. We propose a FGS video codec and a two-layer SNR scalable video codec. Both of these are based on matching pursuits (MP). The bit plane atom coding is presented to achieve the SNR scalable matching pursuits. We also propose a differential quadtree-based algorithm to encode atom positions efficiently. The performance of our methods are compared with those of H.263+ and DCT-based FGS coder.

1. INTRODUCTION

As the development of multimedia application grew, video techniques have been gradually changing from one-to-one to one-to-many communications. In the scalable video coding, a single bitstream is transmitted to all intended recipients. FGS coding has been recently adopted by MPEG-4 to provide the streaming applications based on bit plane coding of discrete cosine transform (DCT) coefficients [1][2].

In lower bit rate video coding, due to a more flexible set of bases in representing residual images and less block artifacts than the block DCT-based transform, the MP-based video codec has better signal-to-noise ratio (PSNR) performance as well as visual quality than that DCT-based video codec [3][4]. SNR-scalable schemes based on matching pursuits have been proposed in [4] [5]. However, neither of them are FGS coding. In [4] the atoms are coded in groups of N atoms at a time to approach the finely scalable codec. However, the number of N should not be too small, for the bit rates will increase rapidly. In this paper, we propose a FGS video codec and a two-layer SNR scalable video codec. Both of them are based on match pursuits. The proposed algorithms use bit plane coding for atom positions and exploits the spatial and temporary dependence between bit planes to encode the atom positions with high coding efficiency.

2. MATCHING PURSUIT AND PROGRESSIVE ATOM CODING

2.1. Matching pursuit

We review the matching pursuit algorithm [6]. Suppose we want to represent a signal $f(x)$ over a redundant dictionary $\mathcal{D}$ with bases $\{g_\gamma(x)\}$. The signal $f(x)$ can be decomposed into

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

where $g_{\gamma_0} \in \mathcal{D}$ and Rf is the residual signal after approximating f in the direction of g_{γ_0} such that $| < f, g_{\gamma_0} > |$ is maximum. The matching pursuit algorithm then decomposes the residual signal Rf by projecting it on a basis function of $\mathcal{D}$, as it was done for f. After M iteration, the signal f can be decomposed into a sum of dictionary elements

$$f = \sum_{k=0}^{M-1} < R^k f, g_{\gamma_k} > g_{\gamma_k} + R^M f. \qquad (2)$$

2.2. Progressive atom coding

When using matching pursuit to represent the motion residual image, the corresponding inner product, basis index and location for each basis is defined as an atom. The coding efficiency of a matching pursuit codec lies in encoding each atom economically. We propose an efficient way to progressively represent the parameters in each atom. They are discussed in the following:

2.2.1. Bit plane encoding of inner product values

We apply the bit-plane based successive-approximation quantization (SAQ) to encode the inner product of atoms. The following algorithm is based on the set partitioning adopted in the famous EZW algorithm, where the atoms of a motion residual frame is organized into $N_{BP} + 1$ bit planes, where N_{BP} corresponds to the most significant bit plane of the atoms.The structure *AtomList* contains the atoms whose inner product values are greater than a threshold given by the current bit plane. The positions, indices of bases, and signs of inner products of the atoms in the list have already been given to a decoder.

0-7803-7622-6/02/$17.00 ©2002 IEEE

Bit Plane Atom Coding Algorithm :

1. **Initialization :** Send N_{BP} to the decoder. Let the index of bit plane n be N_{BP}, and set the $AtomList$ as an empty list.
2. **Refinement Pass :** for each atom in the $AtomList$, output the nth most significant bit of inner product.
3. **Sorting Pass**
   (a) encode the set of atoms, which satisfy $2^n \leq |< R^k f, g_{\gamma_k} >| < 2^{n+1}$.
   (b) move the set of atoms to the end of the $AtomList$.
4. **Next Bit Plane :** decrement n by 1 and go to Step 2.

This algorithm applies the following two phases on each bit plane alternatively: In the refinement phase, the inner product value of each atom in the $AtomList$ are refined by adding one bit information from the current bit plane, while in the sorting phase, new atoms not in the current $AtomList$ are found in the current bit plane and are then appended to the $AtomList$. Although this algorithm explains the encoding on the inner product values, it is also our framework for encoding all the other information of atoms, including their positions, and indices of the basis.

2.2.2. Atom positions encoding with differential quadtree

In a matching pursuit based video codec, the coding performance can be significantly improved if the positions of atoms can be encoded efficiently. We propose a differential algorithm to encode the atom positions. The efficiency of our algorithm lies mainly in the quadtree representation of atom positions and the exploitation of temporary and spatial dependence in the motion residual images.

Quadtree decomposition is a simple technique to represent the atom positions at different resolution levels. The root node of a quadtree represents the entire bit plane. If the entire bit plane has at least one atom, we label the root node with "1"; otherwise, we label it with "0". Four children are then added to the root node, representing four quadrants of the bit plane. This process can be applied recursively to each of the four children until each child node represents a single pixel. Thus, if the image size is $2^{l_{max}} \times 2^{l_{max}}$, the quadtree has at most $l_{max} + 1$ levels. Illustrated in Fig.1 is an example where the node $L(1, 1, 1)$ is the $(1, 1)$ position at level 1 which has four children denoted respectively as $L(0, 2, 2)$, $L(0, 2, 3)$, $L(0, 3, 2)$ and $L(0, 3, 3)$.

The terminal nodes in a quadtree is traversed by a depth-first search algorithm. Both the temporary dependence between the same bit plane in two consecutive frames and the spatial dependence in adjacent bit planes in a frame are used for efficiently encoding the atom positions. The dependence is exploited by the following algorithm based on the differential quadtree in which either the same bit plane in the previous frame or the previous bit plane in the same frame is used to represent the current bit plane. We use $L^t_b(k, i, j)$ to denote a node (k, i, j) in a quadtree corresponding to the b-th bit plane in the t-th residual frame. For the b-th bit plane of the t-th residual frame, our matching pursuit encoder will call the algorithm: Atom Positions Encoding$(l_{max}, 0, 0)$.

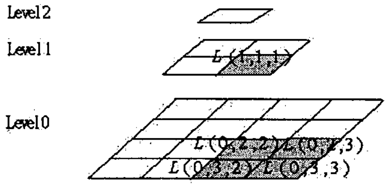


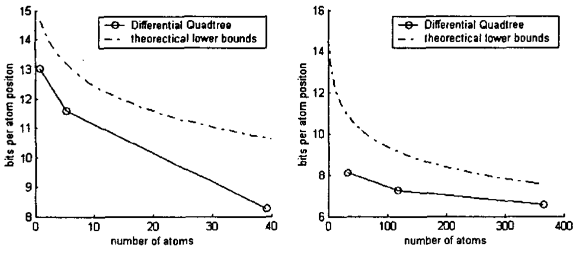
Fig. 1. The Quadtree structure with 3 levels.

```
Atom Positions Encoding $(k, i, j)$
{
  IF $L^{t-1}_b(k, i, j)$ exists
    Output $(L^t_b(k, i, j) \oplus L^{t-1}_b(k, i, j))$;
  ELSE
    Output $(L^t_b(k, i, j) \oplus L^t_{b-1}(k, i, j))$;

  IF $L^t_b(k, i, j) = 0$
    The node is a leaf node;
  ELSE IF ($L^t_b(k, i, j) = 1$ and $k = 0$)
    The node is a leaf node and associates with atom;
    Transmit the index of basis and sign of inner product;
  ELSE encode its four children
    a) Atom Positions Encoding ( k-1, 2i, 2j );
    b) Atom Positions Encoding ( k-1, 2i, 2j+1 );
    c) Atom Positions Encoding ( k-1, 2i+1, 2j );
    d) Atom Positions Encoding ( k-1, 2i+1, 2j+1 );
}
```

If the b-th bit plane is encoded in the previous frame $t-1$, then we encode the EXCLUSIVE OR ($\oplus$) of $L^t_k(i, j)$ and $L^{t-1}_k(i, j)$ to decrease the entropy. Otherwise, the quadtree corresponding to the bit plane $b - 1$ in the current frame t is used to predict the the current quadtree. If a terminal node whose value is 1 is visited, we encode the basis index and the sign of its inner product value.

A theoretical lower bound for the bits in encoding atoms whose positions are assumed to be an independent uniform distribution is given in [4]. Since we have exploited the temporal and spatial dependence of the atoms, our algorithm should produce a better encoding efficiency for atom positions. Moreover, the symbols used in our algorithm are only "0" and "1", thus they are encoded efficiently via the adaptive arithmetic code. In Fig.2 is shown the average number



(a) (b)

Fig. 2. Average number of bits per atom position for each bit plane. (a) Sean. (b) Akiyo.

of bits used to encode one atom position. Our testing images are size of QCIF, encoded at 10 frames/sec, and each frame is encoded with 3 bit planes. The coding efficiency of our algorithm based on different quadtree is evidently superior.

Since the transmitting of differential quadtree is according to the depth-first search, the decoder does not have to receive all the atoms in a particular bit plane in order to construct an image. Any atom received can be decoded immediately. Thus, our algorithm provides fine-grained scalability based on each single atom.

3. FGS MATCHING PURSUIT VIDEO CODING

We propose the FGS matching pursuit video codec as shown in Fig.3. Both the base layer and enhancement layer are based on bit plane atom coding and use differential quadtree for encoding atom positions. The bit plane in enhancement coding is predicted purely from the previous bit plane in the same frame. In this structure, the decoder can stop at any bit rate after it has received the base layer, since the previous enhancement layer is not used to predict the current frame. Fig.4 shows the PSNR comparisons of our MP-based FGS codec with the DCT-based FGS codec. Our implementation of DCT-based FGS is based on the bit plane encoding of full frame DCT coefficients with zig-zag scanning order and is followed by the adaptive arithmetic entropy coding. The testing video in our experiment is the Akiyo sequence, coded at 30 frames/sec. The PSNR is obtained from averaging the first 33 frames in the sequence. To make two video codecs have the same residual image of enhancement layer at each frame. The base layer is reconstructed by the same motion vector without encoding the residual image via DCT or matching pursuit. As one can see, the performance of our MP-based FGS is better than that of DCT-based FGS video codec.

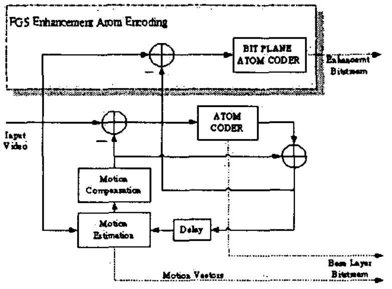


Fig. 3. The FGS Matching Pursuit encoder structure.

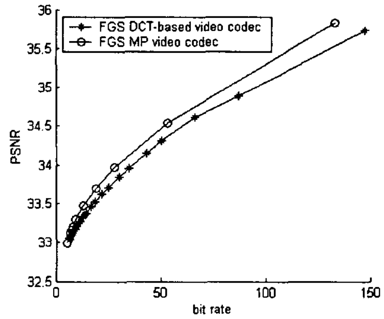


Fig. 4. The PSNR of FGS MP codec compared with FGS DCT-based codec.

4. TWO-LAYER SNR SCALABLE MATCHING PURSUIT CODING

In [4], a two-layer SNR scalable matching pursuit codec is proposed in which the motion vectors computed from the base layer are used to predict the residual images in both the base and enhancement layers. In this section, We propose the two layer SNR-scalable matching pursuit coder as illustrated in Fig.5. Our approach in motion compensation is different from that in [4]. In our approach, the motion vectors used to compensate both layers are estimated from the previous base layer reconstructed images with weight α_1 and from that in the enhancement layer with weight α_2 and $\alpha_1 + \alpha_2 = 1$. The α_1 and α_2 are adjustable parameters according to the percentage of the resource usage by the base layer and the enhancement layer, respectively. The residual images in both layers of our codec are encoded by our algorithm proposed in section 2.2.2. Atoms in both layers are bit-plane encoded. Depending on the bits received, the enhancement layer contains new bit planes, from the refinement of the atoms in base layer and from new atoms not included in the base layer. We compared the two-layer SNR scalable MP codec proposed by us to that in [4]. The two codecs use the same method in encoding residual images

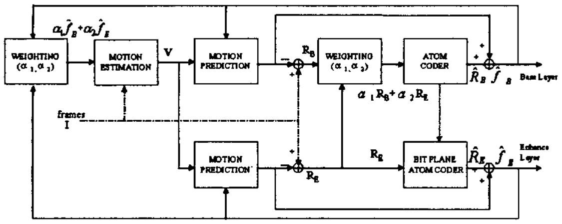


Fig. 5. The two-layer SNR scalable MP structure

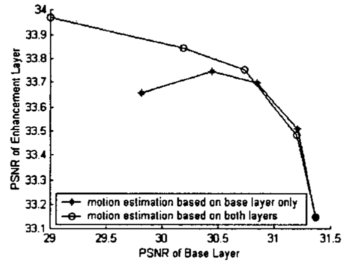


Fig. 6. The PSNR of base layer and enhancement layer with α_1 varying form 0 to 1.

based on bit-plane coding. They are different only in motion vector estimations. The testing image is Sean sequence, coded at 10 frames/sec. In Fig.6, the PSNR obtained in the base layer and the enhancement layer with different methods is shown with α_1 varying form 0 to 1. In all the data, the base layer uses 3 bit plane and the enhancement uses 1 bit plane for both codecs. One can see that the proposed two-layer codec has a better PSNR for the enhancement layer when both codecs are operating with the same PSNR in the base layer for all possible α_1.

We also compared the performance of our two-layer SNR scalable matching pursuit codec with the DCT-based H.263+ coder [7]. The H.263+ coder was implemented by the University of British Columbia. Fig7 illustrates the PSNR of base layer and enhancement layer for our matching pursuit codec and that of H.263+. The performance of ours is better than that of H.263+.

5. CONCLUSIONS

We propose FGS MP video codec and compared its performance to that obtained by DCT-based video codec. We also introduce a two-layer SNR scalable MP codec. In our codec, the motion vectors for motion compensation are estimated from the weighted sum of the motion vectors obtained in both the base layer and enhancement layer. The performance of this codec is compared to those of H.263+ and DCT-based FGS coder and shows PSNR improvement over these. The coding efficiency of our codec is obtained with the combination of bit plane coding and differential quadtree.

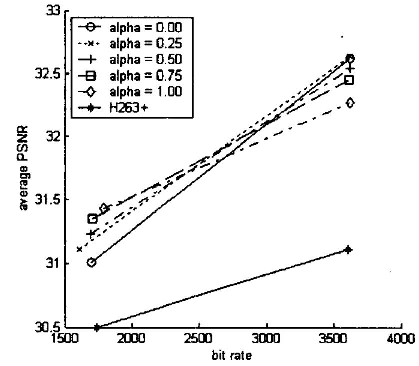


Fig. 7. The PSNR of base layer and enhancement layer for MP and H.263+. alpha = α_1.

6. REFERENCES

[1] H. M. Radha, M. van der Schaar, and Yingwei Chen , "The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP", in *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 53–68, Mar. 2001.

[2] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 301-317, Mar. 2001.

[3] 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.

[4] O. Al-Shaykh, E. Miloslavsky, T. Nomura, R. Neff, and A. Zakhor, "Video compression using matching pursuits", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 123–143, Feb. 1999.

[5] C. D. Vleeschouwer and B. Macq, "SNR scalability based on matching pursuits", *IEEE Trans. on Multimedia*, vol. 2, No. 4, 2000, pp. 198-208.

[6] G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries", *IEEE Trans. Signal Processing*,Vol. 41, pp. 3397–3415, Dec 1993.

[7] ITU-T Recommendation H.263, *Video coding for low bit rate communication*, Feb. 1998.