ATOM POSITION CODING IN A MATCHING PURSUIT BASED VIDEO CODER

*Pierre Garrigues and Avideh Zakhor**

Department of Electrical Engineering and Computer Sciences
University of California at Berkeley, CA 94720
{*garrigue,avz*}@eecs.berkeley.edu

ABSTRACT

In this paper, we propose a new scheme for position coding of the atoms within the Matching Pursuit algorithm as applied to the displaced frame difference (DFD) in a hybrid video encoder. We exploit the spatial and temporal coherence of the positions of the atoms in the DFD: the atoms tend to cluster in regions where the motion prediction fails. The location of these regions is correlated over time. We use a block-based technique in which the number of atoms inside each block is coded differentially with respect to the same block in the previous frame. The atom positions inside each block are coded using the previously proposed NumberSplit algorithm. We demonstrate the effectiveness of our proposed approach on a number of video sequences.

1. INTRODUCTION

The hybrid motion-compensated Discrete Cosine Transform (DCT) structure or a DCT-like integer transform structure are a part of nearly all existing video coding standards such as H.263, H.264/AVC and MPEG-1,2,4, and commercially available video compression systems. In each of these systems, a block-based motion model is used to predict each frame from a previously reconstructed frame, and the residual energy in each motion block is coded using the DCT. One of the problems with DCT is the blocking artifacts typically associated with all block-based transforms. To circumvent blocking artifacts, a number of non-block-based schemes based on expansion of signals on overcomplete set of basis functions have been proposed. One algorithm for such an expansion is Matching Pursuit (MP), originally proposed in the statistical literature [3], and later re-introduced to the signal processing community by Mallat and Zhang [6]. Application of MP to residual video coding was originally proposed in [9], and further developed in [1][2][4][8][5][7]. MP-based codecs have been found to result in significant PSNR and visual quality improvement over DCT-based schemes as they contain no blocking artifacts.

In analyzing the coding statistics of MP-based video codec in [1], we have empirically determined that a significant portion of the bits are devoted to atom position coding. Other sources of bit consumption include motion vectors, atom modulus coding and indices of the elements in the dictionary. Specifically, at higher bit rates, position coding represents up to 40% of the bit rate. Since the performance gap between MP and block-based techniques decreases at higher bit rates, it is conceivable to reduce this gap by more efficient position coding techniques.

In this paper, we propose a new position coding technique based on the exploitation of the spatial and temporal coherence of the atom positions. The outline of the paper is as follows. In Sections 2 and 3, we review the MP theory and existing position coding techniques respectively. Our new position technique and its application are discussed in Section 4. Finally, experimental results are included in Section 5.

2. MATCHING PURSUITS THEORY

The MP algorithm, as proposed by Mallat and Zhang [6], expands a signal onto an overcomplete dictionary of functions. For simplicity, the procedure can be illustrated for a 1-D time signal. Suppose we want to represent a signal $f(t)$ using basis functions from a dictionary set Γ. Individual dictionary functions are denoted as:

$$g_\gamma(t) \in \Gamma$$

where γ is an indexing parameter associated with a particular dictionary element. The decomposition begins by choosing γ to maximize the absolute value of the following inner product:

$$p =< f(t), g_\gamma(t) >$$

We then say that p is an expansion coefficient for the signal onto the dictionary function $g_\gamma(t)$. A residual signal is computed as:

$$R(t) = f(t) - pg_\gamma(t)$$

This residual signal is then expanded in the same way as the original signal. The procedure continues iteratively until either a set number of expansion coefficients are generated

*This work was supported by AFOSR contract F49620-00-1-0327

or some energy threshold for the signal is reached. Each stage n yields an index to the dictionary structure denoted by γ_n, an expansion coefficient p_n, and a residual R_n which is passed on to the next stage. After M stages, the signal can be approximated by a linear function of the dictionary elements:

$$\hat{f}(t) = \sum_{n=1}^{M} p_n g_{\gamma_n}(t)$$

The above technique has useful signal representation properties. For example, the dictionary element chosen at each stage is the element which provides the greatest reduction in mean square error between the true signal $f(t)$ and the coded signal $\hat{f}(t)$. In this sense, the signal content is coded in order of importance, which is desirable in progressive transmission situations, or situations where the bit budget is limited. For image and video coding applications, the most visible features tend to be coded first, while weaker image features are coded later, if at all. The dictionary set we use in this paper consists of an overcomplete collection of 2-D separable Gabor functions [9].

At stage n of the algorithm, the largest inner product p_n, along with the indexing parameter γ_n define an atom. Indexing parameter γ_n consists of (a) the two parameters that define the 2-D separable Gabor functions, and (b) the two position parameters that define its location in the residual image. When the atom decomposition of a single residual frame is computed, it is important to code the resulting parameters efficiently to minimize the resulting bit rate. p_n and the parameters defining the structure of the dictionary are typically coded using fixed Huffman codes. In the remainder of this paper we investigate the coding of the position parameters.

3. ATOM POSITION CODING

We begin this section by deriving an expression for the entropy of independent identically distributed (IID) position parameters of n atoms within a frame of size N pixels. We then provide a brief overview of existing position coding techniques.

3.1. Entropy for IID data

Let us consider the problem of coding the position parameters of n atoms in a frame of size N pixels assuming the positions to be uniformly identically distributed within the frame and independent of each other. We allow several atoms to be at the same position, as it does happen in practice. It can be shown that there are $\binom{N+n-1}{n}$ possible placements of the n atoms within a frame with N pixels, and since the atom positions are IID, they are all equi-probable. Thus the entropy per atom is given by

$$\frac{1}{n}\log_2 \binom{N+n-1}{n}$$

While in practice the atoms for coding the DFD for a video sequence are not IID, the above expression is still a useful reference for atom position coding. Indeed, achieving a lower average bits per atoms is a reasonable indicator of how well we are able to capture the true statistical structure of the atom positions.

3.2. Previous position coding schemes

The position coding technique proposed by Neff and Zakhor [9] is frame based and is based on 2D run-length coding. A differential coding strategy employs three basic codeword tables. The first table P1 is used at the beginning of the frame line to indicate the horizontal distance from the left side of the image to the location of the first atom on the line. For additional atoms on the same line, the second table P2 is used to transmit inter-atom distances. The third table P3 indicates how many lines may be skipped before the next line containing coded atoms. We refer to this position coding scheme as P1P2P3.

A block-based approach was initially introduced in [2]. The method presented in [1] improves upon block-based position coding technique by increasing the coding efficiency without sacrificing error resilience, and works as follows. The atoms within each 16×16 block are first re-ordered according to a spiral scanning order and then coded differentially. Four different Variable Length Coding (VLC) tables are used to Huffman code the run-lengths depending on the number of atoms in it. We refer to this position coding scheme as MB. In [8], the block size is varied from frame to frame using rate optimization.

The NumberSplit algorithm for position coding introduced in [1], is based on divide-and-conquer. First, the total number of atoms T in a given frame is transmitted in the header. Then the image is divided into two halves along a larger dimension, and the number of atoms in the left or top half is coded and transmitted. The algorithm is applied recursively to the halves of the image until there are no more atoms in a given half image or until the size of the half image is one pixel. Note that if we assume that each atom falls uniformly and independently of other atoms onto either half, then the number of atoms in the first half is binomially distributed on $\{0, \cdots, T\}$ with probability of $\frac{1}{2}$. Since the atoms tend to cluster in the DFD where the motion prediction fails, the tails of the binomial distribution are emphasized according to a clustering parameter f. We then use the resulting distribution to build a Huffman table to code the number of atoms in the first half. Hence the NumberSplit algorithm exploits the spatial coherence of the atom positions. We refer to this position coding scheme as NS.

4. OUR PROPOSED POSITION CODING SCHEME

The aim of our proposed scheme is to take advantage of both spatial and temporal coherence of the atom positions. We know that the atoms tend to cluster in regions where the

video sequence	MSE	atoms per frame	IID	P1P2P3		MB		NS		NS+TC		gain	atom positions
				bit-rate	bits/at	bit-rate	bits/at	bit-rate	bits/at	bit-rate	bits/at		
Akiyo	5	151	8.80	32.42	8.54	32.76	8.77	31.66	8.06	30.81	7.49	6.0%	40.3%
	10	86	9.59	20.40	9.54	20.41	9.57	20.00	9.09	19.44	8.59	4.8%	39.9%
	20	47	10.43	12.94	10.59	13.03	10.87	12.69	10.09	12.50	9.71	4.1%	38.5%
Carphone	5	490	7.11	108.94	8.08	114.86	8.82	103.47	7.01	101.14	6.51	11.9%	38.8%
	10	289	7.87	72.94	8.98	72.05	8.54	69.37	7.83	67.88	7.27	5.8%	33.8%
	20	165	8.67	49.02	9.90	47.81	9.20	46.90	8.73	45.99	8.12	3.8%	30.8%
Claire	5	161	8.71	36.27	8.47	36.83	8.78	35.27	7.86	33.97	7.04	7.8%	38.2%
	10	90	9.53	22.85	9.32	22.80	9.25	22.34	8.76	21.61	7.95	5.2%	36.5%
	20	49	10.37	14.62	10.32	14.72	10.58	14.38	9.85	14.14	9.4	3.9%	34.6%
Container	5	384	7.46	73.83	7.76	75.61	8.21	71.05	7.04	69.05	6.51	8.7%	40.1%
	10	207	8.35	42.22	8.50	42.63	8.70	40.86	7.84	39.50	7.18	7.1%	37.6%
	20	114	9.19	25.12	8.96	25.32	9.13	24.68	8.57	23.93	7.91	5.5%	37.6%
Foreman	10	345	7.62	90.11	8.78	90.00	8.59	86.55	7.75	85.12	7.30	5.4%	33.1%
	20	205	8.36	64.10	9.60	62.15	8.86	61.28	8.52	60.28	8.13	3.0%	26.8%
	30	148	8.83	52.50	10.14	51.23	9.20	50.84	9.02	50.10	8.47	2.2%	26.4%
Grandma	5	270	7.97	58.43	8.21	58.99	8.40	56.88	7.65	55.18	7.01	6.5%	38.4%
	10	141	8.89	32.29	9.22	31.94	8.98	31.58	8.73	30.43	7.90	4.7%	39.4%
	20	68	9.92	18.27	10.32	18.15	10.16	17.97	9.87	17.49	9.18	3.6%	37.5%
Hall	5	404	7.39	86.83	7.74	89.11	8.29	84.94	7.27	81.11	6.32	9.0%	37.6%
	10	185	8.51	41.87	8.61	42.18	8.76	40.65	7.95	39.25	7.18	6.9%	33.7%
	30	56	10.19	15.01	9.25	15.45	10.03	14.94	9.11	14.85	8.96	3.9%	33.5%
Highway	5	565	6.90	130.05	7.41	135.79	8.41	127.66	6.99	120.49	5.72	11.3%	34.7%
	10	217	8.28	54.15	8.67	53.88	8.54	52.51	7.92	50.18	6.85	6.9%	34.5%
	20	72	9.84	22.28	10.35	21.92	9.85	21.63	9.45	21.36	9.07	2.6%	32.1%
Salesman	5	300	7.82	64.30	8.31	64.82	8.43	61.96	7.54	60.64	7.07	6.4%	38.9%
	10	170	8.63	38.65	9.16	38.29	8.96	37.41	8.45	36.45	7.87	4.8%	39.3%
	20	93	9.48	24.25	9.92	24.05	9.75	23.69	9.34	23.16	8.78	3.7%	37.2%
Sean	5	238	8.15	49.87	8.10	51.10	8.59	48.18	7.47	47.32	7.09	7.4%	39.4%
	10	140	8.90	30.82	8.92	31.20	9.17	29.95	8.36	29.33	7.91	6.0%	40.4%
	20	82	9.66	20.16	9.81	20.15	9.90	19.68	9.23	19.31	8.84	4.2%	38.7%

Table 1. Comparison of the position coding techniques for various sequences

motion prediction fails. The atom positions are thus spatially highly non-stationary; as such, we take this into account in our coding scheme. Furthermore, from one frame to the next, these regions of high motion are correlated. If there is a large number of atoms in a given region of the DFD, it is likely that there was also a large number of atoms in the same region in the previous frame. In the case of a scalable video coder, a scheme for atom position coding that takes advantage of the temporal coherence was introduced in [5]. The principle is to represent the atom positions by a quadtree and to transmit the XOR of this quadtree with the quadtree of another group of atoms in the previous frame.

We propose a block-based position coding technique where the number of atoms in each block is coded differentially with respect to the same block in the previous frame. If the number of atoms in block b in frame n is $N_{b,n}$, we code the difference $N_{b,n} - N_{b,n-1}$. For the first P frame in the GOP, we code the number of atoms in each block, and for the subsequent P frames we code the difference of numbers of atoms. Thus, at the decoder, if frame $n-1$ has been correctly decoded we have access to $N_{b,n-1}$, and we can recover $N_{b,n}$. In doing so, we take advantage of the temporal coherence. The block size has to be chosen accordingly. Indeed, if the block size is too small, the correlation between $N_{b,n-1}$ and $N_{b,n}$ will be low and it is more efficient to code $N_{b,n}$ directly instead of $N_{b,n} - N_{b,n-1}$. Conversely, if the block size is too large, there are fewer blocks and we achieve lower gain since the number of symbols to code becomes small. We have found 16×16 blocks to be optimal for the video sequences we tested. We use an adaptive arithmetic encoder to code the symbols corresponding to $N_{b,n} - N_{b,n-1}$.

Once we have transmitted the number of atoms, we need to code the atom positions inside each of the 16×16 blocks.

In our video coder the chroma channels are subsampled by a factor of 2 in the horizontal and vertical direction, and thus the positions of the UV atoms are subsampled by 2. In the existing schemes such as [1] the UV atom positions are doubled, and then all the YUV atoms are coded together. This is inefficient since two bits are lost for every UV atom. Our new scheme first subsamples the Y atom positions in the 16×16 block by two, so all the atoms inside that block are defined on an 8×8 grid. We code the Y, U and V atom positions using the NumberSplit algorithm with a clustering parameter set to 0.2 as applied to the 8×8 block. Then, for each atom, we send a codeword specifying if it is a Y, U or V atom, and for the Y atom we add two bits to specify its position in the 16×16 block. Since the atoms tend to lie at the boundaries of the blocks, the NumberSplit algorithm takes into account the non-uniformity of the atom positions inside the blocks. We refer to this position coding scheme as NS+TC.

5. RESULTS

In this Section we present results to compare NS+TC with the frame-based coding technique (P1P2P3), the macro-block based position coding (MB), and the NumberSplit algorithm (NS) described earlier. We use the following QCIF training sequences to compute the frequency tables used to build Huffman tables and for the arithmetic encoder: Silent, Mother, Coast and News. The QCIF test sequences are Akiyo, Carphone, Claire, Container, Foreman, Grandma, Hall, Highway, Salesman and Sean. To approximate constant quality , we encode each frame such that the maximum mean square error (MSE) computed over each macro-block is smaller than a target. We use the values 5, 10 , 20 and 30 as target MSEs. We have found empirically that using values lower than 5 does not offer further visual improvements, and the resulting atoms mostly correspond to noise. The frame rate is 10 Hz, and we code one I frame followed by 99 P frames. We compare the effectiveness of each position coding technique by examining the number of position bits and its impact on the overall bit rate. We also examine the average number of bits per atom for each technique and compare it to the theoretical value, assuming the data is IID. The results are presented in Table 1.

Our proposed scheme NS+TC outperforms all the other schemes in every case. The average number of bits to code the position of an atom is lower than the theoretical value in the case of IID data. For most of the sequences the reduction exceeds 1 bit per atom. This shows that our scheme effectively captures the spatial and temporal coherence. Furthermore, by comparing NS+TC to NS, our scheme uses an average of 0.6 fewer bits. Since NumberSplit exploits the spatial coherence only, this indicates that taking into account the temporal coherence results in a significant gain in coding the atom positions.

The two last columns of Table 1 show the reduction in bit rate when using NS+TC instead of MB, and the percentage of bits used to code the atom positions using MB. Since the percentage of bits used to code the atom positions can exceed 40% of the overall bit rate, there is a significant reduction in the overall bit rate when using a more efficient position coding technique such as NS+TC. We observe that the performance improves with the number of atoms to code. Indeed, at higher bit rates, the percentage of bits allocated to atom coding becomes higher, resulting in more atoms to be coded, and hence larger gain by using a more efficient position coding method. We observe this trend for all of the encoded sequences in Table 1. The lower the MSE target, the larger the reduction in bit rate. The average reduction for MSE values 5, 10 and 20 are 8.3%, 5.8% and 3.8% respectively. The largest gain is obtained with the sequence Highway coded with MSE 5 where the reduction in bit rate is 11.3%. The sequence Foreman has the lowest reduction in bit rate of all sequences, due to its high motion nature and significant percentage of bits going to motion.

6. CONCLUSION

In this paper we have proposed a new scheme to code the positions of the atoms in a Matching Pursuit based video encoder. This new position coding method exploits the statistical structure of the DFD, namely the spatial and temporal coherence. It results up to 11% lower bit rate as compared to existing schemes.

7. REFERENCES

[1] O. Al-Shaykh, E. Miloslavsky, T. Nomura, R. Neff and A. Zakhor, "Video Compression using Matching Pursuits," *IEEE Trans. on Circuits and Systems for Video Tech.*, Vol.9, no.1, pp. 123-143, Feb.1999.

[2] M. Banham and J. Brailean, "A selective update approach to matching pursuits video coding," *IEEE Trans. on Circuits and Systems for Video Tech.*, Vol.7, no.1, pp. 119-129, Feb.1997.

[3] J.H. Friedman and W. Stuetzle, "Projection Pursuit Regression," *Journal of the American Statistical Association*, vol.76, pp.817-823, 1981.

[4] P. Frossard, P. Vandergheynst, R.M. Figueras I Ventura and M. Kunt, "A Posteriori Quantization of Progressive Matching Pursuit Streams," *IEEE Trans. on Signal Processing*, vol.52, no.2, Feb.2004, pp.525-535.

[5] J-L. Lin, W-L. Hwang and S-C. Pei, "SNR Scalability Based on Bitplane Coding of Matching Pursuit Atom s at Low Bit Rates: Fine-Grained and Two-Layer," *IEEE Trans. on Circuits and Systems for Video Tech.*, Vol.15, no.1, Jan.2005.

[6] S. Mallat and Z. Zhang, "Matching Pursuits with time-frequency dictionaries," *IEEE Trans. on Signal Processing*, Vol.41, No.12, pp.3397-3415, Dec.1993.

[7] D.M. Monro and W. Poh, "Improved Coding of Atoms in Matching Pursuits, " *in ICIP 2003*, Barcelona, 2003.

[8] F. Moschetti, K. Sugimoto, S. Kato and M. Etoh, "Bidimensional Dictionary and Coding Scheme for a Very Low Bitrate Matching Pursuit Video Coder," *in ICIP 2004*, Singapore, 2004.

[9] R. Neff and A. Zakhor, "Very Low Bit-Rate Video Coding based on Matching Pursuits," *IEEE Trans. on Circuits and Systems for Video Tech.*, Vol.7, no.1, pp. 158-171, Feb.1997.