Wavelet Lattice Quantization for Low Bit Rate Video Coding

Demetrios G. Sampson, Eduardo A. B. da Silva and Mohammad Ghanbari

University of Essex, Wivenhoe Park, Colchester CO4 3SQ, England

Abstract

A method for low bit rate video coding based on successive approximation wavelet vector quantisation and overlapped block matching motion compensation is described. The main advantage of this scheme is that the most important data of the motion compensated interframe prediction error image are coded with priority. Simulation results are given to evaluate the performance of the coding system at 64 kbit/sec.

1. Introduction

Data compression of video signals is an important research topic with several applications, like video-telephony, video-conferencing and high definition television. Recently, there has been an increasing interest in low bit rate video coding for the transmission of moving pictures through ISDN channels. Several video coding schemes have been proposed to reduce the amount of image data required for transmission. The most successful of these methods aim to remove the temporal redundancies between successive frames of moving sequences by predicting the current frame from the previous ones. This is typically achieved by employing some type of motion estimation/ compensation. The motion compensated interframe prediction (MCIP) error is then coded using a technique, such as transform, subband or vector quantisation.

The ITU-T recommendation H.261 has defined a DCT-based coding scheme operating at integer multiples of 64 kbit/sec, which is suitable for video-phone/video-conferencing applications [1]. Although this video codec is now widely accepted, its picture quality suffers from noticeable blockiness at low bit rates. For this reason, investigations on other video coding methods are conducted aiming to exploit the potential of tools such as wavelet transform and vector quantisation for video coding applications. The success of these techniques is mainly evaluated by the picture quality of the reconstructed frames at low bit rates, where the block transform coders perform poorly.

In this paper we present a method for low bit rate video coding based on wavelet lattice vector quantisation. Motion estimation/compensation for wavelet video coding is first discussed. It is shown that the overlapped block matching (OBM) motion compensation increases the efficiency of the wavelet video codec, by eliminating the blocking artifacts in the prediction error image introduced from the conventional block matching. The motion compensated prediction error signal is coded using a method which combines wavelet transform and lattice vector quantisation, referred to as *Successive Approximation Wavelet Vector Quantization* (SA-W-VQ). In this technique, the most important vectors of wavelet coefficients are successively coded by a series of vectors of decreasing magnitudes. Moreover, the structural similarities among the bands of same orientation are exploited by incorporating a block zero-tree structure. Simulation results demonstrate that the described scheme achieves good coding performance for low bit rate video coding. Comparison with the standard RM8 model of the H.261 video coder [2], shows that the OBM-SAWVQ codec results into improvements in both the peak signal-to-noise ratio performance and the picture quality of the reconstructed image frames.

2. Motion Estimation/Compensation for Wavelet Video Coders

Temporal redundancy between successive image frames can be removed by taking into consideration the displacements of moving objects. Block matching (BM) motion estimation has been widely used in video coding applications. Although BM motion estimation has been proved very successful in reducing the energy (and consequently the amount of data) of the interframe prediction error, it fails to estimate the true motion present in the scene. It can be noticed that irregularities occur in the motion field, so that the motion vectors of neighbouring blocks point towards different directions. As a result of these discontinuities in the motion field, blocking artifacts are introduced into the prediction error image.

Blocking artifacts can have a considerable effect on the coding efficiency of subband/wavelet coders, where the entire image and not a small subimage is transformed. In this case, blockiness on the boundaries of the motion blocks is translated into a large signal power in the high frequency bands. Most subband/wavelet coding techniques tend to spend a large proportion of their bit rate budget just for coding this high frequency information. Hence,

0-7803-2486-2/95 $4.00 © 1995 IEEE

it is important to employ a motion estimation / compensation technique that does not lead to blocking artifacts. In this paper we demonstrate the efficiency of Overlapped Block Matching motion compensation [3] for low bit rate video using wavelet lattice vector quantisation.

In OBM-MC the image frame is partitioned into $2N \times 2N$ overlapped blocks located around a core block of $N \times N$ pixels. Assuming a search area of $\pm s$ pels in the previous frame, the best matched motion vector for each $N \times N$ block is selected by minimising a distortion measure between the $2N \times 2N$ overlapped blocks, after weighting the predicition error with a weighting function $w(x, y)$. In our experiments, we have used the raised-cosine function. The motion compensated image frame is then formed by shifting and weighting each $2N \times 2N$ overlapped block from the previous frame with the estimated displacement vector d_i which corresponds to the $N \times N$ core block indexed by i. The pixel values in the overlapping parts of adjacent blocks are calculated by summing up the weighted pixels.

3. Successive Approximation Lattice Vector Quantisation

An efficient method for coding wavelet transform coefficients has been developed in [4]. According to this method blocks of wavelet coefficients are quantised using a successive approximation vector quantisation (SA-VQ) scheme, such that blocks are coded progressively in several stages. At each stage, the residual quantisation error of the previous passes is further refined, until a certain level of distortion is achieved, or the bit rate budget is exhausted. In this scheme, at each stage the blocks with higher energy are coded first, which ensures that the bits are spend with priority to code the most important information.

A key element of the proposed algorithm is the successive approximation lattice vector quantisation. According to this method, a given input vector x is coded with a series of vectors of decreasing magnitudes $\left\{ \| y_j \| = a^j \| Z \| ; j = 1, 2, \ldots \right\}$, where j is the number of coding stages, $a < 1$ is the *approximation scaling factor*, $\| Z \|$ is larger than the maximum magnitude in the set of input vectors and $\| y_j \|$ is the magnitude of the reconstruction codevector at stage j. At each stage, the orientation of the reconstruction vector is selected from a finite set of unit energy codevectors, referred to as *orientation codebook*, $Y = \left\{ \bar{y} : \|\bar{y}_i\| = 1 ; i = 1, 2, \ldots, N \right\}$, which remains the same at each stage. After n stages, the reconstructed vector is formed as:

$$Q_n(x) = \left\{ a\|Z\|\bar{y}_1 + a^2\|Z\|\bar{y}_2 + \cdots + a^n\|Z\|\bar{y}_n \right\}.$$

In order to guarantee that the residual error $\{ x - Q_n(x) \}$ at stage n, converges into zero as $n \to \infty$, some constraints need to be imposed to the design of the orientation codebook. The number of stages required for almost perfect reconstruction of the original vector depends on the values of : (i) the θ_{max}, which is the maximum angle between any possible input vector and its closest available orientation codevector, and (ii) the approximation scaling factor a. It can be shown [4] that the number of stages increases with both a and θ_{max}. In addition, the larger the value of θ_{max}, the larger is the minimum value of a required for convergence.

The orientation codebooks are designed based on the innermost spherical shells of the regular lattices D_4, E_8 and Λ_{16}, which are shown to offer the best sphere packing properties in their dimensions [5]. The LVQ algorithm described in this section offers the advantage that only a limited number of lattice codevectors are used, so that they can be efficiently encoded using an adaptive arithmetic coder [6]. This avoids the obvious difficulties of designing an efficient entropy coder for a very large lattice codebook, that is typically required in most LVQ methods. Moreover, SA-VQ has a very simple encoding algorithm due to the fast NN algorithms of the lattice quantisation [7].

4. Low bit rate video coding using Wavelet Lattice Quantisation

A video coding method has been developed in [8], which consists of two main parts :
(i) the motion estimation/compensation, where the overlapped block matching algorithm described in section 2 is employed, and
(ii) the wavelet vector quantisation method employed for the compression of the motion compensated interframe prediction error images, which is based on the successive approximation LVQ described in section 3. The flow chart of the coding algorithm is presented in Figure 1.

According to the coding algorithm, the mean value of the MCIP error image is extracted and an M-stage wavelet transform is employed. Each sub-image B_i, where B can be horizontally (H), vertically (V) or diagonally (D) oriented and i=1,2,...,M, is then partitioned to $m \times n$ blocks of wavelet coefficients.

In order to form the input vectors a different scanning is used according to the orientation of the band.

The initial value of the magnitude threshold is set to $T_1 = a\|x\|_{max}$, where a is selected according to the θ_{max} of the orientation lattice codebook and $\|x\|_{max}$ denotes the maximum magnitude of the wavelet coefficient vectors. All vectors in every band are scanned, and the ones with magnitude less than T_1 are marked as zero (*zero blocks*). The rest (*non-zero* or *coded blocks*) are represented by their closest orientation codevector scaled with T_1, that is, $Q_1(x_1) = T_1\tilde{y}_1$. The location of the zero vectors is transmitted using 3 symbols : zero block (Z), zero-tree root (ZT) and coded block (C). Block zerotree roots exploit the similarities among the bands of the same orientation by producing a single symbol to indicate that a block of wavelet coefficients and all its corresponding ones in the higher bands of the same orientation are zero. The arithmetic coder with an adaptive model described in [6] is used to code the string generated by the three symbols (ZT, Z and C). The orientation codevectors for each coded (C) block are also encoded with an arithmetic coder.

The magnitude threshold is updated by multiplying it by a. The non-zero blocks are further refined, by coding the residual error between the original and the reconstructed C blocks with their closest orientation codevector and the new threshold. The indices of the new orientation codevectors are encoded into the bitstream via the arithmetic coder. In the next pass, all zero blocks are scanned again and their magnitudes are compared against the new threshold. A new string of the three symbols is encoded in the bitstream to provide information about the location and the status of the blocks at this stage. As in the previous pass, the indices of the C vectors are coded and the entire process is repeated until a certain bit rate is achieved.

The wavelet coefficient vectors are scanned according to their reconstructed values, the higher energies first, as in [9]. This guarantees that the most important (in terms of energy) image data are always coded first, which is very desirable in video coding, because the bit rate budget is efficiently used for coding those data that would result into maximum distortion with priority. Indeed, an advantage of the described method for video coding applications is that a constant bit rate can be achieved by allocating a fixed number of bits for each frame. This eliminates the need for a buffer to smooth out the bit rate variation.

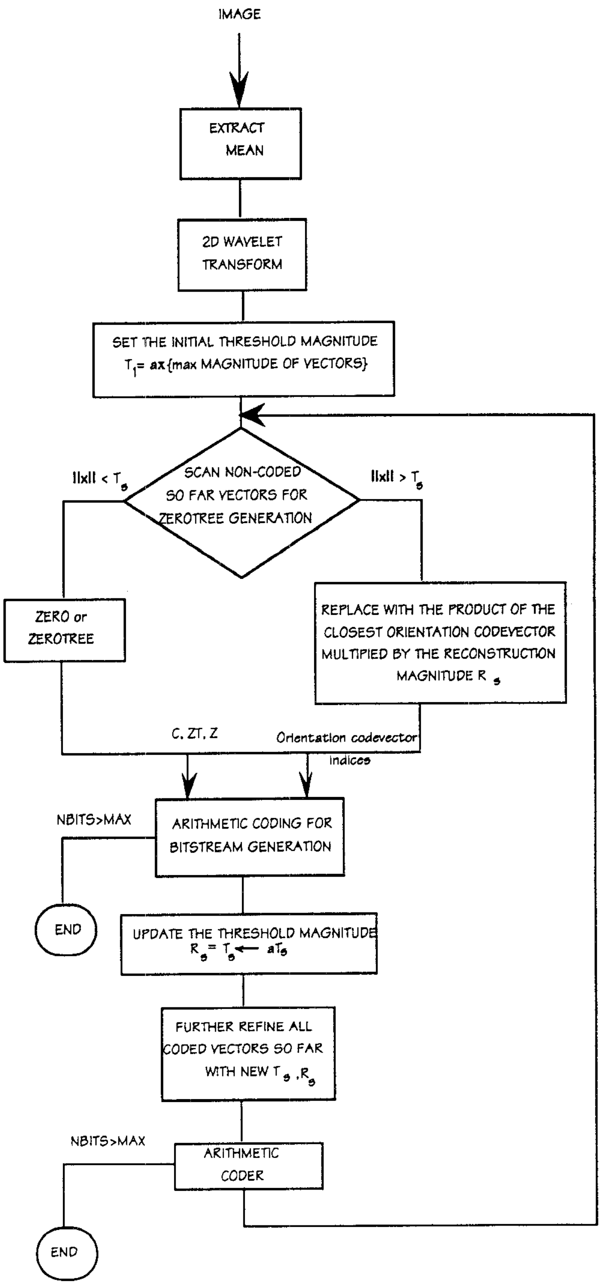

Figure 1: Flow chart of the coding algorithm

5. Experiments - Simulation Results

The performance of the SA-W-VQ for low bit rate video coding applications is evaluated and compared to the RM8 implementation of the standard H.261 video coder and other subband/wavelet schemes reported in literature. For our experiments, we have used the first 100 frames of three test image sequences, namely, MISS AMERICA, CLAIRE and SALESMAN subsampled at 10 Hz.

5.1 Comparison of various lattice codebooks

The performance of various lattice codebooks have been tested. These codebooks are built based on regular lattices which give the best known space-packing properties in dimensions k=4, 8 and 16. Table 1 shows the average luminance peak signal-to-noise ratio (PSNR) achieved by different lattice codebooks for the three test sequences. These results demonstrate that the performance of the coder is robust, i.e., there are only small variations between the various codebooks. However, other factors, such as, the complexity of the encoding algorithm, the population size of the lattice codebook and the vector dimension, can affect the evaluation of the overall performance of the coder. For the rest of the experiments, we have employed the E_8-shell 1+2 lattice codebook, which has given on average the best PSNR performance and it also comprises a reasonable codebook size (N=2160).

LATTICE CODEBOOK	MISS AMERICA	CLAIRE	SALES-MAN
D_4 - shell 1	41.601	39.217	34.130
D_4 - shell 2	41.620	39.492	34.062
D_4 - shell 1+2	41.716	39.373	34.135
E_8 - shell 1	41.472	39.258	33.870
E_8 - shell 2	41.908	39.510	34.118
E_8 - shell 3	41.747	39.477	34.054
E_8 -shell 1+2	41.980	39.594	34.054
E_8-shell1+2+3	41.824	39.310	33.931
Λ_{16}	41.494	39.280	33.787

Table 1. Average Luminance PSNR performance of different lattice codebooks (CIF 10 Hz, 64 kbit/s)

5.2 Overlapped vs conventional Block Matching Motion Compensation

The efficiency of the overlapped block matching motion compensation for the SA-W-VQ video codec is demonstrated in figure 2. In this figure, the PSNR performance of the codec using the OBM and the conventional BM motion compensation, is plotted for coding MISS AMERICA and CLAIRE at 64 kbits/sec. In both cases, the core MC block is 16 × 16 and a search area of ±15 pixels is assumed. Hence, the same number of motion vectors are created. In the OBM-MC, the size of the overlapped blocks is 32 × 32 pixels and the window function is the 2-D raised cosine. In our simulations the absolute values of the horizontal and vertical components of the motion vectors (including non-zero ones) are coded using an adaptive arithmetic coder [6] and embedded into the bitstream. The plots of figure 2 demonstrate the improvement in PSNR obtained by employing the OBM-MC instead of the conventional BM. This improvement is also reflected into the picture quality of the reconstructed frames, following the reasons explained in section 2.

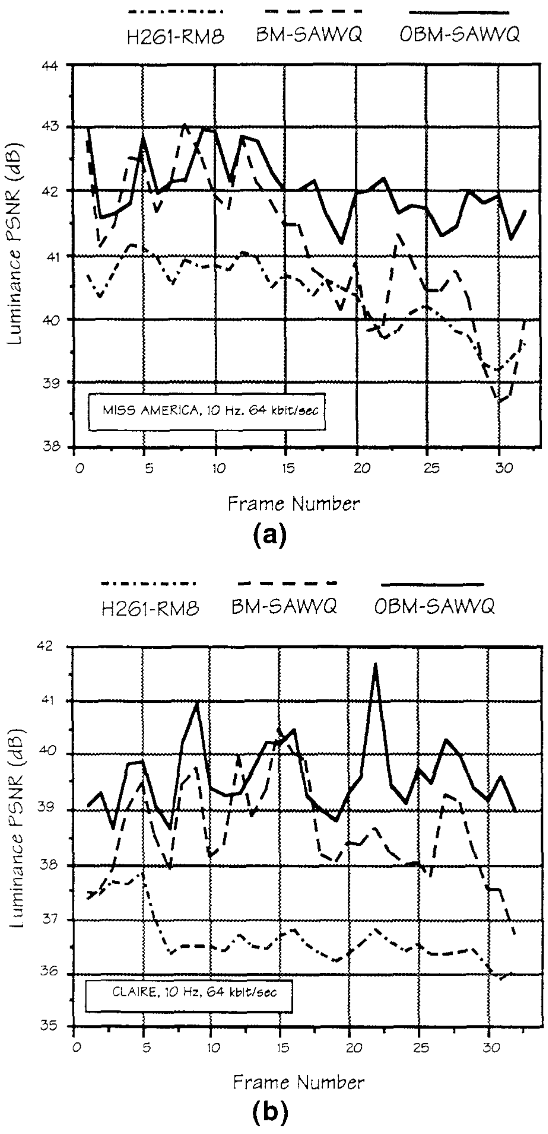

Figure 2 : PSNR performance (a) MISSA AMERICA (b) CLAIRE, CIF, 10 Hz, 64 kbit/sec

5.3 Comparison with other low bit rate video codecs

The performance of the proposed coding scheme is compared with the RM8/H.261 and the overlapped motion compensation/wavelet transform (OMC-WT) developed in [10]. Figure 2 illustrates the PSNR improvement achieved by the OBM-SAWVQ over the RM8 simulations. The picture quality of the OBM-SAWVQ coded pictures is very good and free of the annoying blocking artifacts of RM8 coded images.

The good coding performance presented in Figure 2 and Table 2 is due to an important property of SAWVQ, that is, the most significant information at each image frame is coded with priority, which is a very desirable feature in low bit rate video coding.

IMAGE SEQUENCE	AVERAGE PSNR	METHOD
MISS AMERICA	39.15 dB	OMC-WT [10]
MISS AMERICA	40.33 dB	H.261 -RM8
MISS AMERICA	**41.98 dB**	OBM-SAWVQ
SALESMAN	32.50 dB	OMCWT [10]
SALESMAN	31.90 dB	H.261 -RM8
SALESMAN	**34.05 dB**	OBM-SAWVQ

Table 2 : Average luminance PSNR performance comparison for MISS AMERICA and SALESMAN, CIF 10 Hz, at 64 kbit/s

6. Conclusions

A video coding method based on lattice quantisation of wavelet coefficients has been presented. In this technique, referred as Successive Approximation Wavelet Vector Quantisation (SA-W-VQ), the most important vectors of wavelet coefficients are successively coded by a series of vectors of decreasing magnitudes. Moreover, the structural similarities among the bands of same orientation are exploited by incorporating a block zero-tree structure. We have shown that overlapped block matching motion compensation (OBM-MC) significantly increases the efficiency of the wavelet transform coder, by eliminating the blocking artifacts in the prediction error image introduced from the conventional block matching.

The OBM-SAWVQ video coder offers constant bit rate, with no need for a buffer; yet, remarkably, the PSNR fluctuations from frame to frame are reasonably small. This is due to the fact that the SAWVQ always codes the most important image data first. Moreover, there is small quantisation error accumulation as the image sequence is advanced to higher order frames. Simulation results demonstrate that OBM-SAWVQ achieves improvement both in terms of PSNR and picture quality, compared to the RM8 implementation of the H.261 recommendation.

References

[1] M. Liou, "Overview of the px64 kbit/sec video coding standard," Commun. ACM, vol. 34, pp. 59-63, April 1991.

[2] CCITT SGXV Recommendation Reference Model 8, "Coding for visual telephony," Document 525, June 1989.

[3] R.W. Young and N.G. Kingsbury, "Frequency domain motion estimation using a complex lapped transform," *IEEE Transactions on Image Processing*, vol. 2, no. 1, pp. 2-17, December 1993.

[4] E.A.B. da Silva, D.G. Sampson and M. Ghanbari, "Image coding using successive approximation wavelet vector quantisation," *IEEE Intern. Conf. on Accoustics, Speech and Signal Processing*, ICASSP'95, Detroit, Michigan, May 1995.

[5] J.H. Conway and N.J.A. Sloane, *Sphere Packings, Lattices and Groups*, Springer-Verlag, New York, 1988.

[6] J.W. Witten, R.M. Neal and J.G. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, vol. 30, p. 520-540, June 1987.

[7] D.G. Sampson and M. Ghanbari, "Fast lattice-based gain-shape vector quantization for image sequence coding," *IEE Proceedings Part-I*, vol 140, no. 1, pp. 55-65, February 1993.

[8] D.G. Sampson, E.A.B. da Silva and M. Ghanbari, "A wavelet vector quantisation scheme for image sequence coding at 64 kbit/sec," *Electronic Letters*, vol. 31, no. 2, pp. 92-93, January 1995.

[9] J.M. Shapiro, "Embedded image coding using zerotrees to wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, p. 3445-3462, December 1993.

[10] M. Ohta and S. Nogasi, "Hybrid picture coding with wavelet transform and overlapped motion-compensated interframe prediction coding," *IEEE Transanctions on Signal Processing*, vol. SP-41, no. 12, pp. 3416-3423, December 1993.

Acknowledgement Eduardo A. B. da Silva was supported in part by Universadade Federal do Rio de Janeiro and Conselho Nacional de Desenvolvimento Cientifico e Tecnologico, Brazil, under grant 200885/91-0