Interframe Image Sequence Coding using Overlapped Motion Estimation and Wavelet Lattice Quantisation

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

University of Essex, United Kingdom

ABSTRACT

In this paper we present a method for low bit rate video coding based on wavelet lattice vector quantisation. It is shown that the overlapped block matching (OBM) motion compensation increases the efficiency of the wavelet video codec, by eliminating the blocking artefacts 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 Lattice Vector Quantisation (SA-W-LVQ). In this technique, the most important (in terms of energy) wavelet coefficients are successively coded by a series of vectors of decreasing magnitudes. The structural similarities among the bands of same orientation are exploited by incorporating a block zero-tree structure. Simulation results demonstrate that this scheme achieves very good performance for low bit rate video coding. Comparison with the standard RM8 model of the H.261 video codec, shows that the OBM-SAWLVQ codec results into improvements in both the peak signal-to-noise ratio performance and the subjective quality of the reconstructed pictures.

SUCCESSIVE APPROXIMATION LATTICE VECTOR QUANTISATION

A variation of multi-stage VQ, referred to as *Successive Approximation Vector Quantisation* (SA-VQ) has been proposed in [1]. The basic idea in SA-VQ is that, at each quantisation stage s, any input vector (or residual vector, after the first stage) x_s^i which has energy larger than a given threshold value T_s is represented by a given magnitude R_s (related only to the index of the quantisation stage; but not to the actual energy of the input vector), and an orientation codevector which is selected from the orientation codebook Y to give maximum inner product with x_s^i. The main characteristic of SA-VQ is that it guarantees that the most significant image data are always coded first, so that the available bit rate is spend with priority. There are two elements of information to be transmitted at each stage of the SA-VQ, namely, the index of the selected orientation codevector and the address of the coded vectors.

For a given set of input vectors $\{x^i; i = 1, 2, \ldots, M\}$, the operation of SA-VQ can be described as follows:

$$x_s^i = x_{s-1}^i - SAVQ_{s-1}\left(x_{s-1}^i\right) \longrightarrow \begin{cases} R_s \vec{y}_{sj}, & \text{if } \left\|x_s^i\right\| \geq T_s \\ 0, & \text{if } \left\|x_s^i\right\| < T_s \end{cases}$$

where, x_s^i is the input vector at quantisation stage s with index i; $\left\|x_s^i\right\|$ is the magnitude of x_s^i; T_s is the magnitude threshold at stage s; R_s is the reconstruction magnitude at stage s; $\vec{y}_{sj}$ is the best-matched orientation codevector for x_s^i, selected from the orientation codebook Y, so that :

$$\left(x_s^i \cdot \vec{y}_{sj}\right) \geq \left(x_s^i \cdot \vec{y}_{sn}\right), \quad \vec{y}_{sj}, \vec{y}_{sn} \in Y, \quad j \neq n, \ j, n = 1, 2, \ldots, N$$

Hence, the SA-VQ is designed based on three set of parameters, namely,

(ii) the set of *threshold magnitudes* $\{T_s; s = 1, 2, \ldots\}$;

(ii) the set of *reconstruction magnitudes* $\{R_s; s = 1, 2, \ldots\}$;

(iii the finite set of *orientation codevectors* $Y = \left\{\vec{y}_i : \left\|\vec{y}_i\right\| = 1; i = 1, 2, \ldots, N\right\}$.

Following the description of successive vector approximation illustrated in figure 1, the threshold magnitudes T_s can be selected as $T_s = a^s \left\|x\right\|_{max}$, where $\left\|x\right\|_{max}$ is the maximum magnitude in the set of the original input vectors. The reconstruction magnitudes can be defined, in general, as $R_s = \beta T_s$; however, it is assumed that $\beta = 1$, so that the reconstruction magnitude is equal to the magnitude threshold at any stage s. In this case, the two main design considerations in SA-VQ involve:

Image Processing And Its Applications, 4–6 July 1995
Conference Publication No. 410, © IEE 1995.

(i) the selection of the scaling factor a;

(ii) the selection of the orientation codevectors to be included in the orientation codebook.

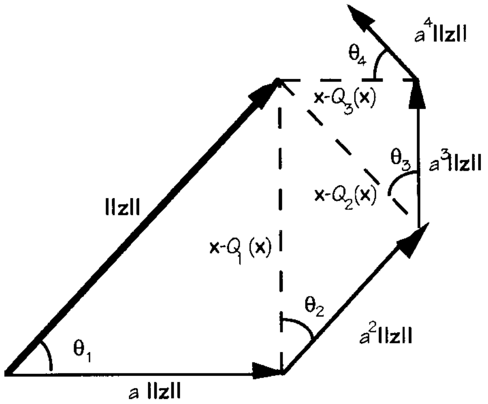


Figure 1 : Successive approximation using vectors

In order to guarantee that the residual error $\{\, x - Q_s(x) \,\}$ at stage s, converges into zero as $s \rightarrow \infty$, 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 [2] 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 [3]. An extra advantage of Successive Approximation Lattice Vector Quantisation (SA-LVQ) is the simplicity of the encoding algorithm due to the fast nearest neighbour algorithms of the lattice vector quantisation [4].

SUCCESSIVE APPROXIMATION WAVELET LATTICE VECTOR QUANTISATION

The success of the Successive Approximation Lattice Vector Quantisation for low bit rate image and video coding applications, depends on the development of efficient methods to encode the two main elements of information in SA-LVQ, namely, the indices of the selected orientation codevectors and the locations of the coded blocks at each quantisation stage. To address these issues, a complete image coding algorithm is described, where SA-LVQ is incorporated with:

(i) the wavelet transform based multi-resolution representation of the image [5];

(ii) the prediction of the non-significant information in different subband images using zero-tree roots [6];

(iii) the entropy coding of both the addressing information and the lattice codevectors indices via adaptive multi-level arithmetic coding [7].

This coding scheme is referred to as *Successive Approximation Wavelet Lattice Vector Quantisation* (SA-W-LVQ). This is an extension of the Embedded Wavelet Zerotree (EWZ) algorithm developed by Shapiro [8] based on successive approximation scalar quantisation (SA-SQ). Figure 2 shows the flow chart of the SA-W-LVQ coding algorithm.

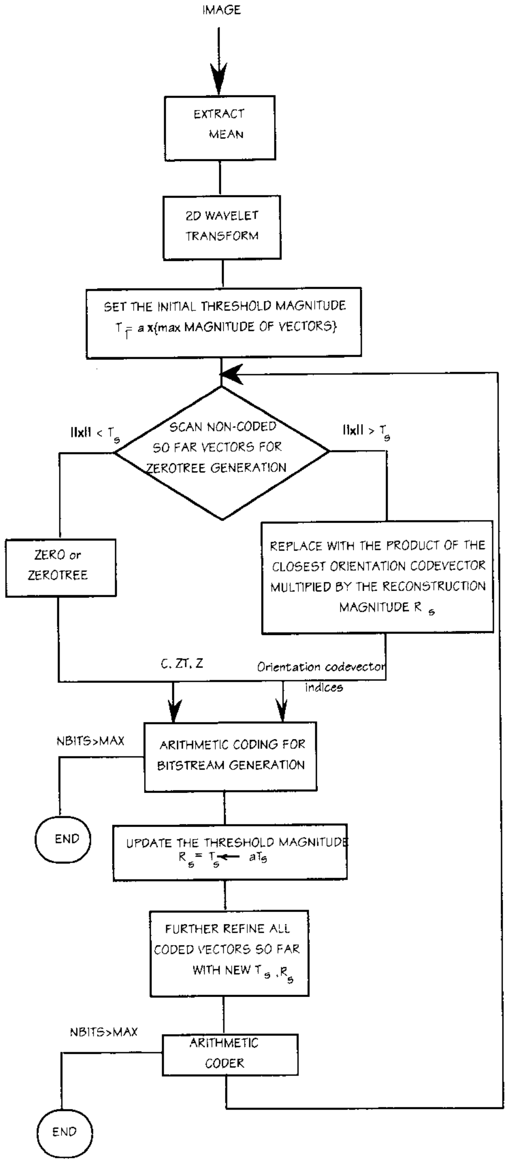


Figure 2: Flow chart of the coding algorithm

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 into $m \times n$ 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 $T_1 = a\|\mathbf{x}\|_{max}$, where a is selected according to the θ_{max} of the orientation lattice codebook and $\|\mathbf{x}\|_{max}$ 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, $Q_1(\mathbf{x}_i) = T_1\vec{\mathbf{y}}_1$. 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 [7] 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.

APPLICATION TO LOW BIT RATE VIDEO CODING

The application of Successive Approximation Wavelet Lattice Vector Quantisation to low bit rate coding of video signals has been investigated [9]. Figure 3 illustrates the block diagram of the encoder.

The two main parts of the codec are:
(i) the motion estimation/compensation, where the overlapped block matching algorithm, 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.

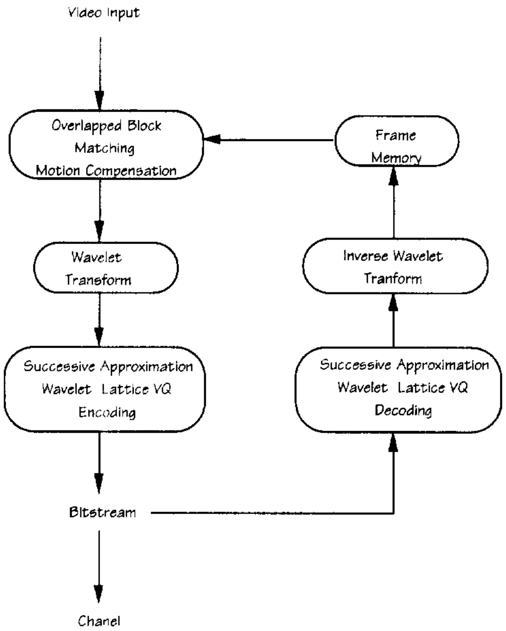


Figure 3 : Block diagram of the OBM-SAWLVQ encoder

Temporal redundancy between successive image frames can be removed by taking into consideration the displacements of moving objects. Although block matching 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. As a result blocking artefacts are introduced into the prediction error image. In case of subband/wavelet codecs, 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, it is important to employ a motion estimation / compensation technique that does not lead to blocking artefacts. Overlapped Block Matching motion compensation (OBM-MC) [10] is tested 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, as shown in figure 4. 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 prediction 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 x 2N overlapped block from the previous frame with the estimated displacement vector $\mathbf{d}_i$ which corresponds to the N x N core block indexed by i. The pixel values in the overlapping parts of adjacent blocks are calculated by summing up the weighted pixels.

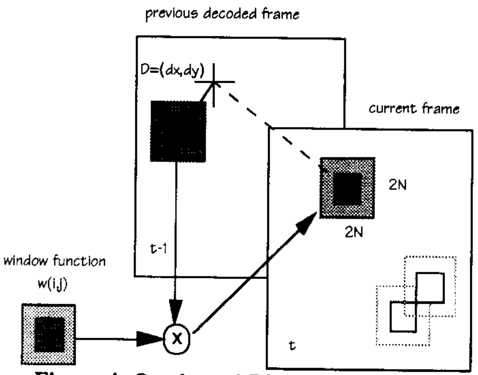


Figure 4: Overlapped Block matching Motion Compensation

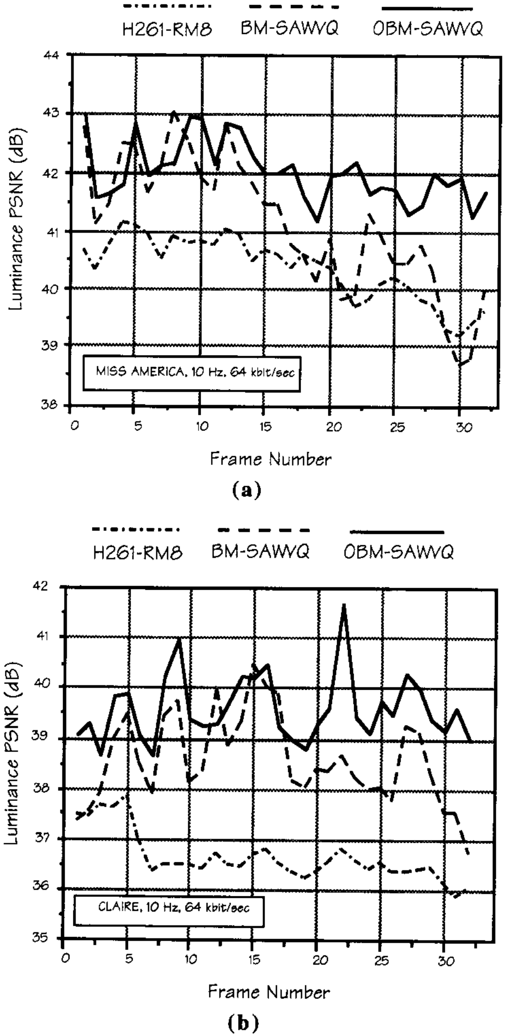
EXPERIMENTS - SIMULATION RESULTS

The performance of the SA-W-LVQ for low bit rate video coding applications is evaluated and compared to the RM8 implementation of the standard H.261 video codec [11]. 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.

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_8shell1+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)

First, 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 k=4, 8 and 16 dimensions. 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).



(a)

(b)

Figure 5 : PSNR performance (a) MISS AMERICA (b) CLAIRE, CIF, 10 Hz, 64 kbit/sec

The efficiency of the overlapped block matching motion compensation for the SA-W-LVQ video codec is demonstrated in figure 5. 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 pixels 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 [7] and embedded into the bitstream. The plots of figure 5 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.

Finally, the performance of the described coding scheme is compared with the RM8/H.261. Figure 5 illustrates the PSNR improvement achieved by the OBM-SAWLVQ over the RM8 simulations. The picture quality of the OBM-SAWLVQ coded pictures is very good and free of the annoying blocking artefacts of RM8 coded images.

CONCLUSIONS

A video coding method based on lattice quantisation of wavelet coefficients has been presented. In this technique, referred as Successive Approximation Wavelet Lattice Vector Quantisation (SA-W-LVQ), 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 artefacts in the prediction error image introduced from the conventional block matching.

The OBM-SAWLVQ 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 SAWLVQ 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-SAWLVQ achieves improvement both in terms of PSNR and picture quality, compared to the RM8 implementation of the H.261 recommendation.

REFERENCES

[1] D.G. Sampson, E.A.B. da Silva and M. Ghanbari, "Wavelet Transform Image Coding using Lattice Vector Quantisation," *Electronic Letters*, vol. 30, no. 18, pp. 1477-1478, September 1994.

[2] 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.

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

[4] 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.

[5] S. G. Mallat, "Multifrequency channel decompositions of images and wavelet models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, volume ASSP-137, pp. 2091-2110, December 1989.

[6] A.S. Lewis and G. Knowles, "Image compression using the 2-d wavelet trasform," *IEEE Transactions on Image Processing*, volume IP-1, pp. 244-250, April 1992.

[7] 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.

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

[9] D.G. Sampson, E.A.B. da Silva and M. Ghanbari, "Wavelet vector quantisation for low bit rate video coding," to appear *IEE Proceedings Vision, Speech and Image Processing* , 1995.

[10] 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.

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

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