# Low bit-rate video coding using wavelet vector quantisation

D.G. Sampson
E.A.B. da Silva
M. Ghanbari

*Indexing terms: Video coding, Lattice vector quantisation, Wavelet transforms*

**Abstract:** A method for low bit-rate video coding based on wavelet vector quantisation is proposed. Motion estimation/compensation using overlapped block matching (OBM) is employed to eliminate the blocking effects in the prediction error introduced by conventional block matching. It is shown that OBM significantly increases the efficiency of the wavelet transform coder. The motion-compensated interframe prediction error is decomposed using a wavelet transform and a method is employed for the efficient coding of the wavelet coefficients. In this technique, the coefficients are coded with a zero-tree multistage lattice vector quantiser. Simulation results are provided to evaluate the coding performance of the described coding scheme for low bit-rate video coding. It provides constant bit rate, obviating the need for buffer, with just small fluctuations in PSNR. Moreover, comparison with the RM8 implementation of the standard H261 video coder shows that the presented codec provides improvements in both peak signal-to-noise ratio and picture quality.

## 1 Introduction

Data compression of video signals is an important research topic with several applications, such as videotelephony, videoconferencing and high-definition television (HDTV). Several video coding schemes have been proposed to reduce the amount of image data required for transmission and storage. 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, sub-band or vector quantisation.

The ITU recommendation H261 has defined a coding scheme operating at integer multiples of 64 kbit/s, which is suitable for videophone/videoconferencing applications [1]. This is a hybrid DPCM/DCT coder, where the MCIP error image is partitioned into blocks of $8 \times 8$ pixels which are transformed using a 2-D DCT. The transform coefficients are quantised with a uniform scalar quantiser and are encoded with a variable-length coder. 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 [2–11]. 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.

A wavelet transform is the decomposition of a signal into expansions and translations of a mother function $\psi(t)$, and it has been used in various image processing and computer vision applications [12]. The wavelet transform provides a tool for decomposing images into multiresolution approximations, which correspond to different frequency components of the original image. This can be implemented via an octave-band sub-band analysis/synthesis process, leading to a tree-structured hierarchical decomposition as shown in Fig. 1. Several

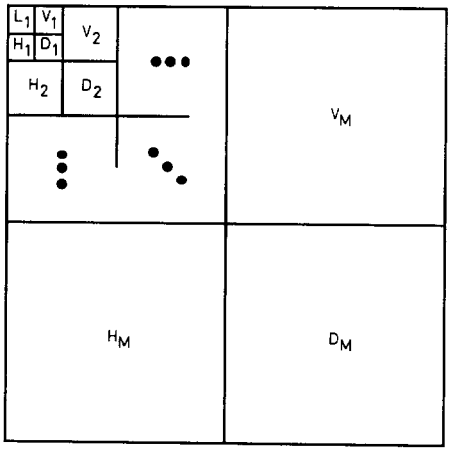



**Fig. 1** *Two-dimensional M-stage wavelet transform*

investigations have been reported regarding the development of efficient methods for coding wavelet coefficients [13–16]. Typically, the most successful of these methods



Paper 1920K (E4), first received 5th September 1994 and in revised form 27th February 1995

The authors are with the Department of Electronic Systems Engineering, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, United Kingdom

Mr. 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.

take into consideration the structural similarities between bands of the same orientation by generating zero tree roots.

Vector quantisation (VQ) is a generalisation of scalar quantisation (SQ), where a group of samples are jointly quantised instead of separate quantisation of each individual sample. The theoretical advantage of VQ over SQ has motivated a plethora of investigations on VQ as a data compression method [17]. In practice, the most important feature of VQ is the potential of achieving fractional bit allocation for the vector components. Lattice vector quantisation (LVQ) is a type of VQ where the codebook is designed based on regular lattices. A regular lattice is a discrete set of points in $k$-dimensional space, which can be generated by the integral linear combination of a given set of basis vectors [18]. The main advantage of LVQ is the significant reduction in the encoding complexity typically required by full-search VQ [19]. This is accomplished by exploiting the structural properties of regular lattices.

In this paper we propose a method for low bit-rate video coding based on wavelet lattice vector quantisation. Motion estimation/compensation for wavelet video coder is first discussed. It is shown that the overlapped block matching (OBM) proposed in Reference 20 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 motion-compensated interframe prediction error signal is coded using a method which combines wavelet transform and lattice vector quantisation, referred to as successive approximation wavelet vector quantization (SAWVQ). The basic principles of SAWVQ and the evaluation of its performance for still-image coding are described in Reference 21. 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 the same orientation are exploited by incorporating a block zero-tree structure. Simulation results are provided to evaluate the coding performance of the described coding scheme for low bit-rate video coding. Moreover, comparison with the RM8 implementation of the standard H261 video coder [22] shows that the presented codec provides improvements in both the peak signal-to-noise ratio and the picture quality of the reconstructed frames.

## 2 Motion estimation/compensation for wavelet video codecs

Temporal redundancy between successive image frames can be removed by taking into consideration the displacements of moving objects. Thus, motion estimation/compensation is an important part of any video coder [23]. Block matching (BM) motion estimation has been widely used in video coding applications and fast implementations have been suggested [24].

In the conventional BM algorithms, the image is partitioned into nonoverlapping, rectangular blocks. Each block in the current frame is then compared against all possible blocks of same dimensions within a specified search area in the previous frame. The best match is found either by minimising a distortion cost function (i.e. the mean square error, MSE) or by maximising a correlation function (i.e. the crosscorrelation) between the two blocks. The estimated block motion vector $\boldsymbol{d} = (dx, dy)$ indicates the displacement of the current block in relation to the previous frame.

Although BM motion estimation has 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. Irregularities occur in the motion field, such that the motion vectors of neighbouring blocks point towards different directions. As a result of these discontinuities in the motion field, considerable blocking artifacts are introduced into the prediction error image. Fig. 2*a* illustrates

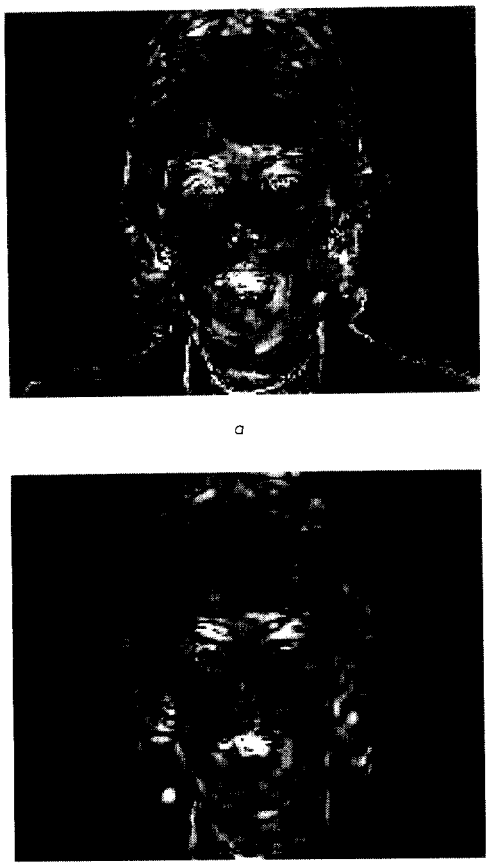


**Fig. 2** *Zoomed part of prediction error signal between frames 1 and 4 of 'Claire'*

*a* Conventional block matching
*b* Overlapped block matching

an example of the prediction error image obtained after conventional block-matching motion compensation. It can be observed that the boundaries of adjacent blocks are noticeable.

For block transform coders, possible blocking artifacts present in the prediction error image do not significantly reduce the coding performance, provided that the motion compensation block dimensions are multiples of the transform block ones. On the other hand, blockiness 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, blocking artifacts on the boundaries of the motion blocks is translated

into a large signal power in the high-frequency bands. Most sub-band/wavelet coding techniques will tend to spend a large proportion of their bit rate budget just for coding this high-frequency information which is not required since it is not part of the original scene.

Hence, it is important to employ a motion-estimation/compensation technique that does not lead to blocking artifacts. Possible candidates are pel-recursive [23] or phase corelation [25]. In this paper we have used a variation of block matching, where the blocks are overlapped with each other, so called overlapped block matching (OBM) motion compensation. Overlapped motion compensation methods have been proposed in References 20 and 26, in connection to lapped orthogonal transforms. Ohta and Nogasi [9] have recently used OBM in their wavelet transform video coder. Here, we demonstrate the efficiency of OBM-MC for low bit-rate video using SAWVQ.

The operation of the OBM-MC is illustrated in Fig. 3. The image frame is partitioned into $2N \times 2N$ overlapped

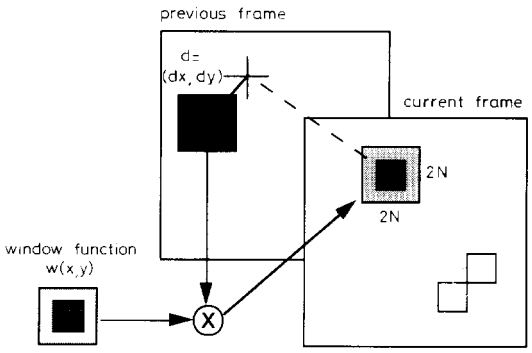



**Fig. 3** *Overlapped block matching motion compensation*

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 prediction error with a weighting function $w(x, y)$. In our experiments, we have used the raised-cosine function given by

$$w(x, y) = \cos^2\left(\frac{x\pi}{2N}\right)\cos^2\left(\frac{y\pi}{2N}\right)$$

where $x, y = -N, \ldots, N$ (1)

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 $\boldsymbol{d}_i$ 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 the weighted pixels. Fig. 2*b* illustrates a magnified part of the prediction error image obtained by OBM. It is evident, by comparing the images in Fig. 2*a* and 2*b*, that the MCIP error image created by OBM-MC is free from any blocking artifacts. In Section 5, we demonstrate that this can lead to significant improvement of the coding performance achieved by the proposed wavelet video coder.

## 3 Successive approximation lattice vector quantisation

We have developed an efficient method for coding wavelet transform coefficients [21, 27]. In this method blocks of wavelet coefficients are quantised with a successive approximation vector quantisation (SAVQ) scheme, such that blocks are coded progressively in several stages. At each stage, the residual quantisation errors of the previous passes are further refined until a certain level of distortion is achieved, or the bit-rate budget is exhausted. At each stage the blocks with higher energy are coded first, which ensures that the bits are spent 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 $\boldsymbol{x}$ is coded with a series of vectors of decreasing magnitudes $\{\|\boldsymbol{y}_j\| = a^j\|z\|;\ j = 1, 2, \ldots\}$, where $j$ is the number of coding stages, $a < 1$ is the approximation scaling factor, $\|\boldsymbol{y}_j\|$ is the magnitude of the reconstruction codevector at stage $j$, and $\|Z\|$ is larger than the maximum magnitude of the set of input vectors. At each stage, the orientation of the reconstruction vector is selected from a finite set of unit energy codevectors, referred to as orientation codebook, $Y = \{\boldsymbol{y} : \|\boldsymbol{y}_i\| = 1;\ i = 1, 2, \ldots, N\}$, which remains the same at each stage. This is illustrated in Fig. 4. After $n$

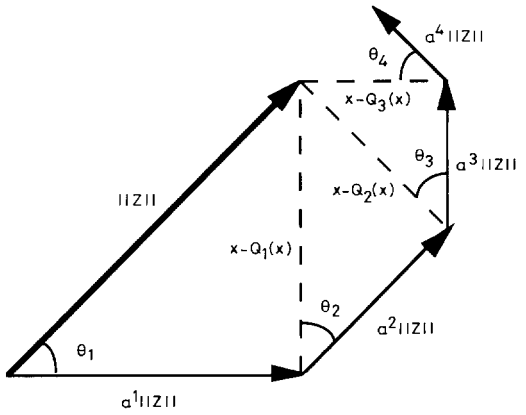



**Fig. 4** *Successive approximation using vectors*

stages the reconstructed vector is formed as

$$\boldsymbol{Q}_n(\boldsymbol{x}) = \{a\|Z\|\boldsymbol{y}_1 + a^2\|Z\|\boldsymbol{y}_2 + \cdots + a^n\|Z\|\boldsymbol{y}_n\} \quad (2)$$

To guarantee that the residual error $\{\boldsymbol{x} - \boldsymbol{Q}_n(\boldsymbol{x})\}$ 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: (*a*) the $\theta_{max}$, which is the maximum angle between any possible input vector and its closest available orientation codevector, and (*b*) the approximation scaling factor $a$. The sufficient conditions to guarantee the convergence of the successive approximation by a finite set of orientation vectors of decreasing lengths have been derived by considering the worst case. This can be evaluated by assuming that the maximum error in orientation occurs at every stage, i.e. $\theta_i = \theta_{max}$, $i = 1, 2, \ldots$. From Fig. 4, the error made after the first pass has a magnitude given by

$$\|\boldsymbol{r}_1\|^2 = \|\boldsymbol{x} - \boldsymbol{Q}_1(\boldsymbol{x})\|^2 = \|\boldsymbol{z}\|^2 + a^2\|\boldsymbol{z}\|^2 - 2\|\boldsymbol{z}\|a\|\boldsymbol{z}\|\cos\theta_{max} \quad (3)$$

After $n$ passes the magnitude of the residual vector is given by

$$\|r_n\|^2 = \|r_{n-1}\|^2 + a^{2n}\|z\|^2 - 2\|r_{n-1}\|a^n\|z\| \cos\theta_{max} \quad (4)$$

The residual vector magnitudes after each pass $\|r_i\|$, $i = 1, 2, \ldots, n$ can be calculated for any given pair $(a, \theta_{max})$. The recursive formula is used to calculate the convergence scaling factor $\bar{a}$, for any $\theta_{max} \in [0°, 90°)$ such that the scheme converges for any $a \geq \bar{a}$, where $a \in [0.5, 1)$. Fig. 5*a* gives the values of the convergence scaling factor $\bar{a}$ for all angles $\theta \in [0°, 90°)$, and Fig. 5*b* plots $\theta_{max}$ against the number of iterations required for convergence when $a = \bar{a}$.

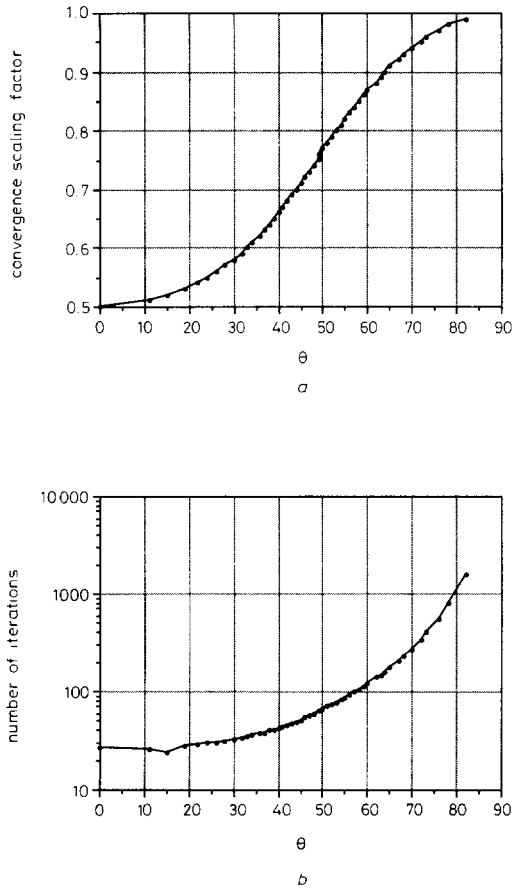



**Fig. 5** *Graphs of θ against*
*a* Convergence scaling factor $\bar{a}$
*b* Number of iterations required for convergence

From Fig. 5, the number of stages increases with both $a$ and $\theta_{max}$. In addition, the higher the value of $\theta_{max}$, the higher the minimum value of $a$ required for convergence. To achieve optimum compression ratios, the number of stages required to obtain a certain distortion level must be minimum. Thus, it is implied that the selected orientation codebook must be designed so that $\theta_{max}$ is as small as possible. Nevertheless, there is a compromise between the value of $\theta_{max}$ and the resolution of the orientation codebook, determined by $\log_2 N/k$, where $N$ is the codebook population and $k$ is the vector dimension. In the scalar case ($k = 1$), $\theta_{max} = 0°$. However, bigger gains in bit-rate reduction can be achieved, when higher dimensional vectors are used, despite the larger values of $\theta_{max}$, owing to the fractional bit allocation obtained by vector quantisation.

We have designed orientation codebooks based on regular lattices. An important category of lattices are the root lattices, namely, $Z_k(k > 1)$, $A_k(k > 1)$, $D_k(k > 3)$, $E_k(k = 6, 7, 8)$ and the Barnes–Wall $\Lambda_{16}$, which have been shown to offer the best sphere-packing properties in their dimensions [18]. In general, the points of a given regular lattice $L_k$ are distributed on the surface of successive, concentric, $k$-dimensional hypershells $S_m$, centred at the origin, so that all lattice points at the same shell have the same $l_r$-norm. The following parameters are important for the evaluation of the efficiency of a particular lattice-orientation codebook, assuming that it is built by the lattice points from $S_m$ shell of a given lattice $L_k$:

(*a*) solid angle between nearest neighbour codevectors $\theta_{NN}(L_k, S_m)$

(*b*) population of lattice points on the particular shell $N_m(L_k, S_m)$

(*c*) maximum possible angle between any input vector and its closest codevector

$$\theta_{max}(L_k, S_m)$$

Table 1 summarises the parameters of regular lattices that give the best lattice packing at dimensions $k = 4$, 8 and 16.

**Table 1: Parameters of regular lattices with best packing at dimensions $k = 4, 8, 16$**

| Lattice type, $L_k$ | Shell index, $m$ | Population $N_m$ | Nearest neighbour angle, $\theta_{NN}$ | Maximum angle, $\theta_{max}$ |
|---|---|---|---|---|
| $D_4$ | 1 | 24 | 60° | 45° |
| $D_4$ | 2 | 24 | 60° | 45° |
| $D_4$ | 1 + 2 | 48 | 45° | 32° |
| $E_8$ | 1 | 240 | 60° | 45° |
| $E_8$ | 2 | 2160 | 41° | 45° |
| $E_8$ | 3 | 6720 | 33° | 35° |
| $E_8$ | 1 + 2 | 2400 | 41°/45° | 32° |
| $E_8$ | 1 + 2 + 3 | 9120 | 30°/35° | 29° |
| $\Lambda_{16}$ | 2 | 4320 | 60° | 55° |

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 [28]. This avoids the obvious difficulties of designing an efficient entropy coder for a very large lattice codebook, which is typically required in most LVQ methods. There are some similarities between the SAVQ and multistage gain/shape VQ [17]. However, in SAVQ there is no need to code the values of the gain, since the reconstruction magnitude is fixed and known at each stage. Finally, SAVQ has a very simple encoding algorithm due to the fast NN algorithms of the lattice quantisation [19].

## 4 Low bit-rate video coding using wavelet lattice quantisation

This Section describes a method suitable for low bit-rate coding of video signals. The proposed coder consists of two main parts:

- motion estimation/compensation, where the overlapped block matching algorithm described in Section 2 is employed, and
- 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 basic principles of the coding algorithm are outlined in Fig. 6. According to this algorithm, the mean value of the MCIP error image is extracted and an $M$-stage

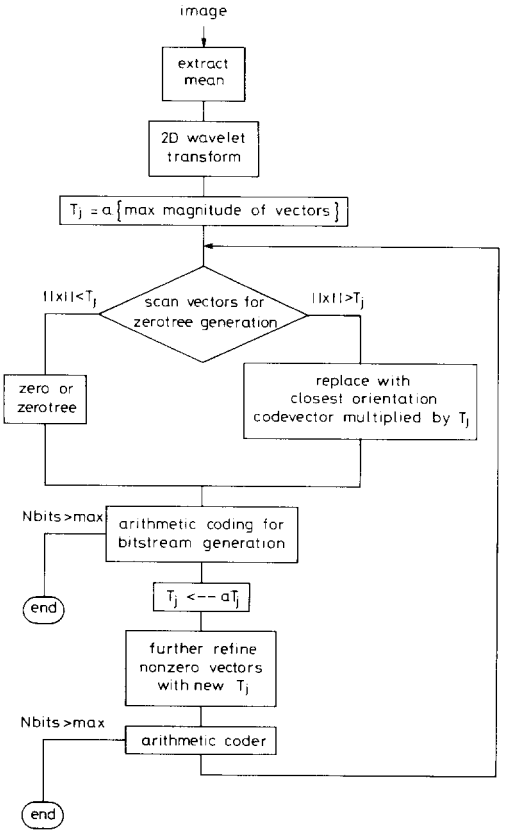



**Fig. 6** *Flow chart of coding algorithm*

wavelet transform is employed, resulting into an image decomposition as in Fig. 1. Each subimage $B_i$, where $B$ can be horizontally (H), vertically (V) or diagonally (D) oriented and $i = 1, 2, \ldots, M$, is then partitioned to $m \times n$ blocks of wavelet coefficients. To form the input vectors a different scanning is used according to the orientation of the band [4].

After scanning, the algorithm proceeds as follows. The initial value of the magnitude threshold is set to $T_1 = a\|x\|_{max}$, where $a$ is selected according to the $\theta_{max}$ 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 (nonzero or coded blocks) are represented by their closest orientation codevector scaled with $T_1$, that is $Q_1(x_i) = T_1 y_1$. The location of the zero vectors is transmitted using three symbols: zero block (Z), zero-tree root (ZT) and coded block (C). Block zero-tree 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 Reference 28 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 then updated by multiplying it by $a$. The nonzero 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 Reference 15. This guarantees that the most important information is always coded first, which is very desirable in video coding. Indeed, an important advantage of the proposed coding algorithm 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 of a buffer for smoothing out the bit-rate variation. Coding the most important coefficients (in terms of energy) always first guarantees that the bit-rate budget will be efficiently used for coding those image data that would result into maximum distortion.

Successive refinement of the wavelet coefficients offers control over the maximum level of quantisation error made in each coefficient. This can be important for reducing unpleasant artifacts in picture quality, since the error introduced by a poorly quantised single-wavelet coefficient can be spread over an area of the reconstructed image [29]. Furthermore, it provides the means to guarantee that an arbitrary level of average distortion for each band is met [30]. This could be convenient for performing bit allocation among the wavelet coefficients taking into consideration the human visual system (HVS) sensitivity to each frequency band.

## 5 Experiments: simulation results

In this Section, the performance of the successive approximation wavelet vector quantisation (SAWVQ) for low bit-rate video coding applications is evaluated and compared to the RM8 implementation of the standard H261 video coder and other sub-band/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'. The original pictures are in CIF format with frame rate 30 frame/s. We have used 10 frame/s for all experiments at 64 kbit/s by just subsampling the original sequence by a factor of three in the temporal domain, without any extra processing. The 2-D wavelet transform used in this work is an octave-band decomposition, which is implemented by the biorthogonal filter bank with coefficients shown in Table 2. In all experiments using SAWVQ, the first frame of the sequence is coded using intraframe SAWVQ at 0.5 bpp.

### 5.1 *Selection of number of stages of wavelet decomposition*

First, we investigated the effect of the number of stages of the wavelet transform in the coding performance. Table 3 shows the average luminance peak signal-to-noise ratio (PSNR) for all three test image sequences using 2–5 stage

**Table 2: Coefficients of filter bank used**

| | $H_0(z)$ | $G_0(z)$ | $H_1(z)$ | $G_1(z)$ |
|---|---|---|---|---|
| $z^7$ | 0.0000 | 0.0000 | −0.0059 | 0.0000 |
| $z^6$ | 0.0000 | −0.0059 | 0.0033 | 0.0000 |
| $z^5$ | 0.0200 | −0.0033 | 0.0360 | 0.0000 |
| $z^4$ | −0.0111 | 0.0360 | −0.0097 | −0.0200 |
| $z^3$ | −1.1426 | 0.0097 | −0.0723 | −0.0111 |
| $z^2$ | 0.0448 | −0.0723 | −0.2223 | 0.1426 |
| $z^1$ | 0.5889 | 0.2223 | 0.8135 | 0.0448 |
| $z^0$ | 0.5889 | 0.8135 | −0.8135 | −0.5889 |
| $z^{-1}$ | 0.0448 | 0.8135 | 0.2223 | 0.5889 |
| $z^{-2}$ | 0.1426 | 0.2223 | 0.0723 | −0.0448 |
| $z^{-3}$ | −0.0111 | −0.0723 | 0.0097 | −0.1426 |
| $z^{-4}$ | 0.0200 | 0.0097 | −0.0360 | 0.0111 |
| $z^{-5}$ | 0.0000 | 0.0360 | −0.0033 | 0.0200 |
| $z^{-6}$ | 0.0000 | −0.0033 | 0.0059 | 0.0000 |
| $z^{-7}$ | 0.0000 | −0.0059 | 0.0000 | 0.0000 |

wavelet transform. For this experiment, the $\Lambda_{16}$-based lattice codebook is employed. A 256 × 256 window of the original CIF pictures is used, to facilitate the implementation of multistage decomposition. The results reported in Table 3 demonstrate that there are no significant differences between the various decomposition stages. Nevertheless, the two-stage one gives better performance for all three test sequences. Hence, we have used a two-stage wavelet transform for the rest of the simulations, since it can be directly implemented with the original CIF image frames, and it also has a faster implementation than the others.

**Table 3: Average luminance PSNR performance of different number of stages of wavelet transform**

| Stages | 'Miss America' 10 Hz, 64 kbit/s | 'Claire' 10 Hz, 64 kbit/s | 'Salesman' 10 Hz, 64 kbit/s |
|---|---|---|---|
| 2 | 41.21 | 39.00 | 33.72 |
| 3 | 40.98 | 38.82 | 33.54 |
| 4 | 41.08 | 38.64 | 33.35 |
| 5 | 41.04 | 38.86 | 33.06 |

### 5.2 Comparison between 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 gives the parameters of these lattices. In our experiments, the best value of the approximation scaling factor $a$ was estimated for the different lattice codebooks from the results obtained by coding the first frame (intraframe mode).

Table 4 shows the average luminance PSNR achieved by different lattice codebooks for the three test sequences.

**Table 4: Average luminance PSNR performance of different lattice codebooks**

| Lattice codebook | 'Miss America' CIF 10 Hz, 64 kb/s | 'Claire' CIF 10 Hz, 64 kb/s | 'Salesman' CIF 10 Hz, 64 kb/s |
|---|---|---|---|
| $D_4$-shell 1 | 41.60 | 39.21 | 34.13 |
| $D_4$-shell 2 | 41.62 | 39.49 | 34.06 |
| $D_4$-shell 1 + 2 | 41.71 | 39.37 | 34.13 |
| $E_8$-shell 1 | 41.47 | 39.25 | 33.87 |
| $E_8$-shell 2 | 41.90 | 39.51 | 34.11 |
| $E_8$-shell 3 | 41.74 | 39.47 | 34.05 |
| $E_8$-shell 1 + 2 | 41.98 | 39.59 | 34.05 |
| $E_8$-shell 1 + 2 + 3 | 41.82 | 39.31 | 33.93 |
| $\Delta_{16}$ | 41.49 | 39.28 | 33.78 |

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 effect 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 ($N = 2160$).

### 5.3 Comparison between overlapped and conventional block-matching motion compensation

The efficiency of the overlapped block-matching motion compensation (OBM-MC), described in Section 2, for the SAWVQ video coder is demonstrated in Fig. 7. The

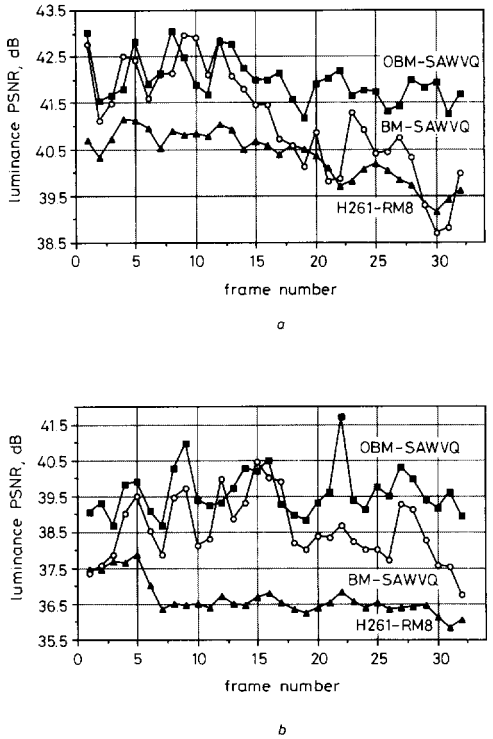



**Fig. 7** *Performance comparison between overlapped and conventional block matching*

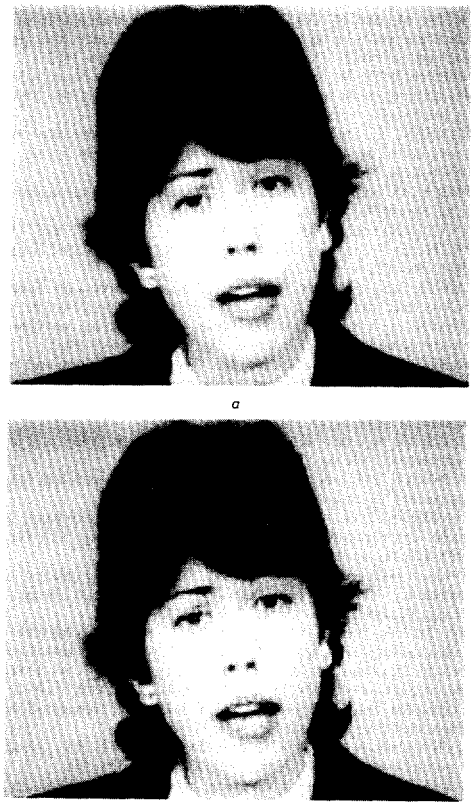
*a* 'Miss America'. CIF 10 Hz, 64 kbit/s
*b* 'Claire'. CIF 10 Hz, 64 kbit/s
—▲— H261-RM8
—○— BM-SAWVQ
—■— OBM-SAWVQ

figure plots the PSNR performance of the coder using the OBM and the conventional BM motion compensation, for coding 'Miss America' and 'Claire' at 64 kbit/s. 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 given by eqn. 1. In our simulations the absolute values of the horizontal and vertical components of the motion vectors (including zero ones) are coded using an adaptive arithmetic coder [28] and embedded into the bitstream. The plots of Fig. 7 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, for the reasons explained in Section 2. It is interesting that, in the case of OBM, as opposed to the conventional BM, there is no

PSNR degradation as the sequence advances towards higher order frames.

### 5.4 Comparison with other low bit-rate video coders

Finally, the performance of the proposed coding scheme is compared with the RM8 implementation of the rec-



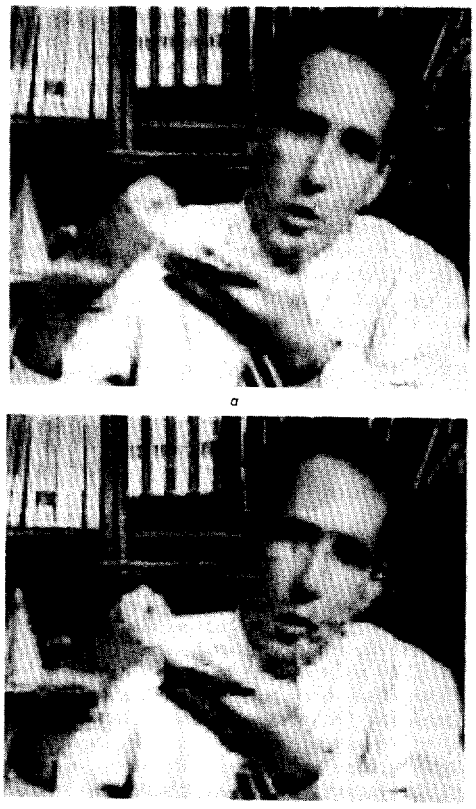
a

b

**Fig. 8** *Zoomed part of reconstructed frame 73 'Claire', CIF 10 Hz, 64 kbit/s*

*a* SAWVQ
*b* H261/RM8

ommendation H261 and other video coders developed for low bit-rate video coding applications. Fig. 7 illustrates the PSNR improvement achieved by the OBM-SAWVQ over the RM8 simulations. The picture quality of the reconstructed frames is demonstrated in Fig. 8. Fig. 8*a* and *b* show a zoomed part of frame 73 of 'Claire'



a

b

**Fig. 9** *Zoomed part of reconstructed frame 43 'Salesman', CIF 10 Hz, 64 kbit/s*

*a* SAWVQ
*b* H261/RM8

coded by OBM-SAWVQ and RM8, respectively, at 64 kbit/s. A magnified part of frame 43 of 'Salesman', also coded at 64 kbit/s using OBM-SAWVQ and RM8, is shown in Figs. 9*a* and *b*. It is evident that the picture quality of the OBM-SAWVQ coded pictures is very good

**Table 5: Performance comparison with other low bit-rate video coders**

| Image sequence | Spatial res. | Frame rate | Bit rate | Average Lum PSNR | Method | Ref. |
|---|---|---|---|---|---|---|
| | | Hz | kbit/s | dB | | |
| Miss America | 256 × 256 | 10 | 64 | 38.10 | sub-band/VQ | 8 |
| Miss America | 256 × 256 | 10 | 64 | 41.80 | OBM-SAWVQ | |
| Miss America | 352 × 288 | 10 | 700 | 39.00 | ECSBC/AEC | 7 |
| Miss America | 384 × 288 | 10 | 332 | 40.52 | zerotree | 33 |
| Miss America | 352 × 288 | 10 | 260 | 38.63 | edge sub/VQ | 10 |
| Miss America | 352 × 288 | 10 | 100 | 39.45 | 3D-WT | 31 |
| Miss America | 352 × 288 | 10 | 64 | 39.15 | OBM-WT | 9 |
| Miss America | 352 × 288 | 10 | 64 | 40.33 | H261-RM8 | |
| Miss America | 352 × 288 | 10 | 64 | 41.98 | OBM-SAWVQ | |
| Claire | 256 × 256 | 10 | 360 | 36.00 | WT/adaptVQ | 32 |
| Claire | 256 × 256 | 10 | 64 | 39.35 | OBM-SAWVQ | |
| Claire | 352 × 288 | 10 | 64 | 36.64 | H261-RM8 | |
| Claire | 352 × 288 | 10 | 64 | 39.59 | OBM-SAWVQ | |
| Salesman | 352 × 288 | 10 | 64 | 32.50 | OBM-WT | 9 |
| Salesman | 352 × 288 | 10 | 64 | 31.90 | H261-RM8 | |
| Salesman | 352 × 288 | 10 | 64 | 34.05 | OBM-SAWVQ | |
| Salesman | 352 × 288 | 15 | 700 | 33.90 | edge sub/VQ | 10 |

and free of the annoying blocking effects of RM8 coded images.

Table 5 shows the average PSNR obtained by various low bit-rate subband/wavelet video coders reported in literature. These include sub-band/residual VQ [8], edge-based sub-band VQ [10], 3D wavelet transform with adaptive bit-plane run-length coding [31], entropy-constained sub-band combined with hierarchical motion compensation [7], adaptive wavelet VQ [32], overlapped motion estimation with entropy-coded wavelet transform [9] and a zero-tree subband method [33]. We have also included the average luminance PSNR obtained by the RM8 implementation of H261 Recommendation [22] and the proposed OBM-SAWVQ.

The good results presented in Fig. 7 and Table 5 are due to an important property of SAWVQ, that is, the most significant information at each image frame is coded prioritarily, which is a very desirable feature in low bit-rate video coding. Indeed, the effectiveness of this approach can be seen not only on the high values of the averge PSNR achieved, but also on the relatively small variations in PSNR obtained with exactly constant bit rate, without a buffer.

## 6 Conclusions

We have proposed a video coding method based on lattice quantisation of wavelet coefficients. In this technique (SAWVQ) 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 presented offers constant bit rate, with no need for a buffer; yet, remarkably, the PSNR fluctuations from frame to frame are reasonably small for the image sequences that have been tested. 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 advances to higher order frames. Simulation results for three standard test image sequences, namely, 'Miss America', 'Claire' and 'Salesman', demonstrate that OBM-SAWVQ achieves promising performance.

## 7 References


1 LIOU, M.: 'Overview of the p × 64 kbit/sec video coding standard', *Commun. ACM*, April 1991, **34**, pp. 59–63

2 WOODS, J.W., and NAVEEN, T.: 'Subband encoding of video sequences', *Proc. SPIE*, Visual Communications and Image Processing IV, 1989, **1999**, pp. 724–732

3 WESTERINK, P.H., BIEMOND, J., and MULLER, F.: 'Subband coding of image sequences at low bit rates', *Signal Processing: Image Communication*, 1990, **2**, pp. 441–448

4 GHARAVI, H.: 'Subband coding of video signals', *in* WOODS, J.W. (Ed.): 'Subband image coding' (Kluwer, 1991), chap. 6

5 ZHANG, Y.-Q., and ZAFAR, S.: 'Motion-compensated wavelet transform coding for color video compression', *IEEE Trans.*, 1992, **CSVT-2**, (3), pp. 285–296

6 DUFAUX, F., MOCCAGATTA, I., ROUCHOUZE, B., EBRAHIMI, T., and KUNT, M.: 'Motion-compensated generic coding of video based on multiresolution data structure', *Opt. Eng.*, 1993, **32**, (7), pp. 1559–1570

7 KIM, Y.-H., and MODESTINO, J.W.: 'Adaptive entropy-coded subband coding of image sequences', *IEEE Trans.*, 1993, **COM-41**, (6), pp. 975–987

8 MERSEREAU, R.M., SMITH, M.T.J., KIM, C.S., KOSSENTINI, F., and TRUONG, K.K.: 'Vector quantisation for video data compression', *in* SEZAN, M.I., and LAGENDIJK, R.L. (Eds): 'Motion analysis and image sequence processing' (Kluwer, 1993), chap. 9, pp. 257–285

9 OHTA, M., and NOGASI, S.: 'Hybrid picture coding with wavelet transform and overlapped motion-compensated interframe prediction coding', *IEEE Trans.*, 1993, **SP-41**, (12), pp. 3416–3423

10 MOHSENIAN, N., and NASRABADI, N.M.: 'Edge-based subband VQ techniques for images and video', *IEEE Trans.*, 1994, **CSVT-4**, (1), pp. 53–67

11 NAVEEN, T., and WOODS, J.W.: 'Motion compensated multiresolution transmission of high definition video', *IEEE Trans.*, 1994, **CSVT-4**, (1), pp. 29–41

12 RIOUL, O., and VETTERLI, M.: 'Wavelet and signal processing', *IEEE Signal Process. Mag.*, 1991, **8**, (4), pp. 14–38

13 ANTONINI, M., BARLAUD, M., MATHIEU, P., and DAUBECHIES, J.: 'Image coding using wavelet transform', *IEEE Trans.*, 1992, **IP-1**, (2), pp. 205–220

14 LEWIS, A.S., and KNOWLES, G.: 'Image compression using the 2-D wavelet transform', *IEEE Trans.*, 1992, **IP-1**, pp. 244–250

15 SHAPIRO, J.M.: 'Embedded image coding using zerotrees to wavelet coefficients', *IEEE Trans.*, 1993, **41**, pp. 3445–3462

16 MOCCAGATA, I., and KUNT, M.: 'Vector quantisation and cross-bands prediction for color image coding'. Presented at the international symposium on *Picture coding*, PCS-94, Sacramento, CA, September 1994, paper 14.05

17 GERSHO, A., and GRAY, R.M.: 'Vector quantisation and signal compression' (Kluwer, New York, 1991)

18 CONWAY, J.H., and SLOANE, N.J.A.: 'Sphere packings, lattices and groups' (Springer-Verlag, New York, 1988)

19 SAMPSON, D.G., and GHANBARI, M.: 'Fast lattice-based gain-shape vector quantization for image sequence coding', *IEE Proc. I*, 1993, **140**, (1), pp. 55–65

20 WATANABE, H., and SINGHAL, S.: 'Windowed motion compensation', *Proc. SPIE*, Visual Communications and Image Processing '91, 1991, **1605**, pp. 582–589

21 DA SILVA, E.A.B., SAMPSON, D.G., and GHANBARI, M.: 'Image coding using successive approximation wavelet vector quantization'. Presented at IEEE international conference on *Acoustics, Speech, Signal Processing*, ICASSP-95, May 1995, paper TP01.1

22 'Coding for visual telephony'. CCITT SGXV Recommendation reference model 8. document 525, June 1989

23 MUSMANN, H.G., PIRSCH, P., and GRALLERT, H.J.: 'Advances in picture coding', *Proc. IEEE*, 1985, **73**, (4), pp. 523–548

24 GHANBARI, M.: 'The cross-search algorithm for motion estimation', *IEEE Trans.*, 1990, **COM-38**, (7), pp. 950–953

25 AHMAD FADZIL, M.H., and DENNIS, T.J.: 'Video subband VQ coding at 64 kbit/s using short-kernel filter banks with an improved motion estimation technique', *Signal Processing: Image Communication*, 1991, **3**, pp. 3–21

26 YOUNG, R.W., and KINGSBURY, N.G.: 'Frequency domain motion estimation using a complex lapped transform', *IEEE Trans.*, 1993, **IP-2**, (1), pp. 2–17

27 SAMPSON, D.G., DA SILVA, E.A.B., and GHANBARI, M.: 'Wavelet transform image coding using lattice vector quantisation', *Electron. Lett.*, 1994, **30**, (18), pp. 1477–1478

28 WITTEN, J.W., NEAL, R.M., and CLEARY, J.G.: 'Arithmetic coding for data compression', *Communications of the ACM*, 1987, **30**, pp. 520–540

29 MACQ, B.: 'Weighted optimum bit allocation to orthogonal transforms for picture coding', *IEEE J. Sel. Areas Commun.*, 1992, **10**, pp. 875–883

30 DA SILVA, E.A.B., and GHANBARI, M.: 'Linear phase wavelet transforms for low bit rate image coding'. 7th European Signal Processing Conference, EUSIPCO-94, Edinburgh, September 1994, p. 6B.3

31 YAO, S., and CLARKE, R.J.: 'Image sequence coding using adaptive vector quantisation in wavelet transform domain', *Electron. Lett.*, 1992, **28**, (17), pp. 1566–1568

32 GOH, K.H., SORAGHAN, J.J., and DURRANI, T.S.: 'New 3-D wavelet transform coding algorithm for image sequences', *Electron. Lett.*, 1993, **29**, (4), pp. 401–402

33 BHUTANI, G., and PEARLMAN, W.A.: 'Image sequence coding using the zero-tree method', *Proc. SPIE*, Visual Communications and Image Processing '93. Vol. 2094, Cambridge, MA, November 1993, pp. 463–471