ADVANCED MOTION THREADING FOR 3D WAVELET VIDEO CODING*

Lin Luo[1], Feng Wu[2], Shipeng Li[2], Zixiang Xiong[3], and Zhenquan Zhuang[4]

[1]IBM China Research Laboratory, Beijing, 100085
[2]Microsoft Research Asia, Beijing, 100080
[3]Dept of Electrical Engineering, Texas A&M University, College Station, TX 77843
[4]University of Science and Technology of China, Hefei, China

ABSTRACT

This paper presents an advanced motion threading technique for improved performance in 3D wavelet coding. First, we extend an original motion threading idea of ours to a lifting-based implementation. Methods for processing many-to-one pixel mapping and non-referred pixels and for enabling fractional-pixel alignment in motion threading are proposed to reduce the wavelet boundary effects. Second, we devise an advanced motion threading technique, in which one set of motion vectors is generated for each temporal layer of wavelet coefficients for temporal scalability. In order to reduce the motion overhead information, especially at low bit rates, several correlated motion prediction modes at the macroblock level are defined to exploit the intra/inter layer correlation in motion vector coding. Finally, rate-distortion optimization is utilized in motion estimation to select the best motion prediction mode for each macroblock. With the new motion threading technique, we are able to achieve 1.5-6.0 dB gain in average PSNR in 3D wavelet coding over our previous implementation of motion threading.

Keywords: 3D wavelet coding, layered video coding, lifting-based wavelet transform, motion threading, motion estimation, motion vector coding, rate-distortion optimization.

*Part of this work was presented at ICME'01 [14], VCIP'03 [24], and ICIP'03 [25].

1. INTRODUCTION

The wavelet transform [1] provides a multi-scale representation of images and video in the space-frequency domain. Aside from the energy compaction and decorrelation properties that facilitate compression, a major advantage of the wavelet transform is its inherent scalability. The wavelet-based JPEG2000 standard [2] not only gives superior compression performance over the DCT-based JPEG standard, but also offers scalabilities in rate, quality and resolution that are very desirable for consumer and network applications.

3D wavelet video coding [3-25], on the other hand, was only recently shown to be able to compete in performance with conventional MC-DCT based standard approaches (e.g., H.264). It is believed that there is still a long way to go before 3D wavelet video coding becomes the industry norm in practical applications. The main reason is because motion estimation/compensation in standard video coding fully takes advantage of a locally homogeneous motion model, which conveniently captures the local motion characteristics of many real-world video sequences. With the global wavelet transform that is typically applied to the whole image, however, it is not easy to accommodate a local motion model. Work on wavelet-based video coding so far can be roughly divided into two classes: the 2D wavelet transform plus motion compensation (2D+MC) [26-30] and 3D wavelet coding.

With 2D+MC [26-30], one method is to directly apply the 2D wavelet transform to the motion-compensated residual frame. Since MC effectively reduces cross-frame redundancies, pixels in a residual frame are much less correlated than in the original frames. The wavelet transform is less effective in representing residual frames in terms of decorrelation. Thus the 2D+MC method does not offer noticeable improvement over standard MC-DCT approaches. Another 2D+MC method is to apply the 2D wavelet transform to each original video frame before performing motion compensation in the wavelet domain. Due to the fact that the wavelet transform employed in 2D+MC is often critically sampled, shift variance associated with these transforms limits the effectiveness of MC. For example, a single pixel shift in the image domain will result in two very different wavelet image representations, making it difficult for MC to work well. Hence this method cannot achieve high coding efficiency either.

In 3D wavelet video coding (e.g., [10-12]), a temporal-domain wavelet transform is usually applied before 2D wavelet transform on each resulting video frame. With a 3D wavelet representation, it is relatively easy to achieve coding scalability in rate, quality and resolution. However, due to object motion in a scene, spatially co-located pixels across frames are usually misaligned. Thus, in order to take full advantage of the temporal-domain wavelet transform and hence achieve high coding efficiency, wavelet filtering has to be performed along the motion trajectories. Due to the complications involved in coupling the wavelet transform with motion estimation, the bottleneck in efficient 3D wavelet video coding lies in motion estimation.

There have been a number of attempts to incorporate motion estimation into 3D wavelet video coding. Taubman *et. al* [5] pre-distorted the input video sequence by translating frames relative to one another before the wavelet transform so as to compensate for camera pan motion. Wang *et. al* [8] used the mosaic technique to warp each video frame into a common coordinate system and applied a shape-adaptive 3D wavelet transform on the warped video. Both of these schemes adopt a global motion model that is inadequate for enhancing the temporal correlation among video frames in many sequences with local motion. To overcome the limitation of a global motion model, Ohm [4] proposed a block matching technique that is similar to the one used in standard video coders while paying special attention to covered/uncovered and connected/unconnected regions. Failing to achieve perfect reconstruction with motion alignment at half-pixel resolution, Ohm's scheme does not show significant performance improvement.

Several groups have looked into combining motion compensation with the lifting-based wavelet transform since 2001 [13-25]. One noteworthy work is [13], in which the authors

implemented the first sub-pixel (½-pel) resolution motion estimation scheme within the motion-compensated lifting framework. They were also the first to incorporated overlapped block motion estimation into the lifting-based temporal wavelet transform. However, the Haar filters were used in [13] and the authors erroneously stated that "the coding efficiency is not increased significantly by performing a temporal analysis with longer filters". Luo *et. al* [14] first employed the 5/3 biorthogonal wavelet filters with ½-pel resolution motion estimation in 2001. Later in the same year, Secker and Taubman used both the Haar and 5/3 filters (again with ½-pel resolution motion estimation, see [16] and its conference version). Several follow-up works (e.g., [18, 22]) also demonstrated the advantage of the 5/3 filters for temporal wavelet transformation. Other related publications with different focuses (e.g, using the 9/7 or longer filters, ¼-pel motion estimation, adaptive or low-complexity implementation, scalable motion coding, etc.) appeared in [17-25].

Recently, MPEG is actively exploring and promoting inter-frame wavelet coding techniques. The MC-EZBC coder [17, 18] proposed by Chen and Woods has become well known because of its good performance. In MC-EZBC [17], each pair of frames is motion estimated in a hierarchical fashion to achieve up to 1/8-pel accuracy before going through a motion-aligned lifting-based Haar transform. Additional wavelet decompositions along the temporal direction are performed on the lowpass frames by following the same procedure. Using sub-pel motion alignment, MC-EZBC [17] performs about the same as the new H.264 JVT standard [31] for some sequences. In the original MC-EZBC coder [17], the Haar filters do not fully exploit the long-term correlation across video frames. In addition, motion vectors at each temporal decomposition layer are estimated and coded independently without exploiting the cross-layer correlations among them. In an improved version of MC-EZBC [18], the 5/3 filters are used in conjunction with improved motion vector coding.

In [11], Xu *et. al* proposed a motion threading (MT) approach that employs longer wavelet filters to exploit the long-term correlation across frames along the motion trajectory. The baseline architecture of the original 3D MT wavelet coder of [11] is shown in Figure 1, where each column represents a frame. Backward motion estimation is performed on each pair of frames from $Frame_0$ to $Frame_{n-1}$ at the macroblock level. Pixels along the same motion trajectory are linked to form a non-overlapping thread according to the motion vectors of the macroblocks they belong to. The biorthogonal 9/7 or 5/3 wavelet filters are used for transforming the threads before each resulting frame going through a 2-D spatial wavelet transform with the 9/7 filters. After the 3D wavelet transform, all wavelet coefficients are coded into one embedded bitstream.

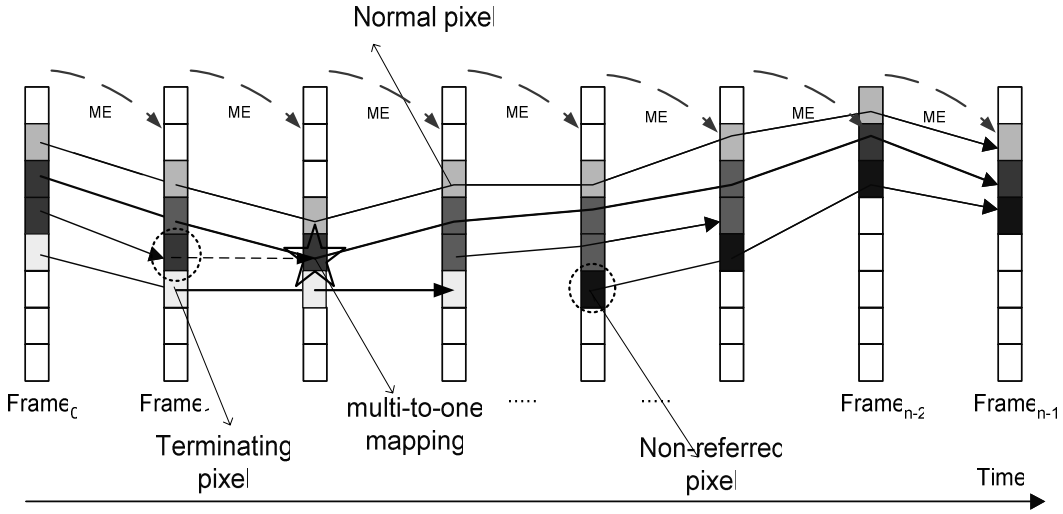


Figure 1: The original MT implementation in [11].

Motion threading offers a flexible yet efficient way of incorporating motion alignment into the temporal-domain wavelet transform, with encouraging results reported in[11]. However, MT still has limitations. On one hand, motion threads are not allowed to overlap so as to guarantee perfect reconstruction at the decoder side. If multiple pixels in the current frame are mapped to the same pixel in the next frame, i.e., whenever a scenario of *many-to-one mapping* occurs, only one of them in the current frame can be linked to the one in the next frame. Each of the remaining pixels has to be marked as a terminating pixel to signify that its associated thread ends at the current frame. On the other hand, for pixels in the current frame that are not linked by those in the previous frame, i.e., when there are *non-referred pixels*, we have to start a new motion thread for each of them. When the input sequence has complex motion, the number of many-to-one mappings or non-referred pixels (or thread boundaries) will be large. Due to the boundary effects in wavelet reconstruction [12], these artificial thread boundaries will degrade the coding performance.

Motion threading is only performed at full-pel resolution in [11]. It is well known that motion estimation at ½- or ¼-pel resolution can significantly improve the coding efficiency in standard video coding [16, 17, 18]. We thus aim to extend MT to sub-pel resolution in this paper (as was done in [14]) while maintaining perfect reconstruction for improved 3D wavelet video coding. We first adopt a lifting-based wavelet transform for the motion threads. We carefully manage cases with many-to-one pixel mapping and non-referred pixels in MT in order to reduce the number of thread boundary pixels (hence the boundary effects). We propose techniques for ½- or ¼-pel motion estimation and motion alignment under the perfect reconstruction constraint [24]. We also generate one set of motion vectors for each temporal layer of wavelet coefficients for temporal scalability. Since motion vectors in adjacent regions within a temporal layer and the same region in different layers have strong correlations, we devise a correlated motion estimation (CME) scheme to facilitate motion vector coding [25]. Eight modes are designed to capture the variation of local motion correlation at the macroblock level and a rate-distortion (R-D) optimized algorithm is employed to select the best mode for each macroblock.

Compared to [15], CME in [25] shares the same idea of exploiting the correlation among cross-layer motion vectors. However, the two approaches are different: CME reduces the number of motion vectors by defining eight modes, whereas the scheme in [15] aims at improving the prediction accuracy in motion vector coding. Thus, although our CME technique is designed for MT, the idea equally applies to other 3D wavelet coding schemes, such as MC-EZBC [18] and those in [15, 16].

The rest of the paper is organized as follows. Section 2 describes the proposed advanced lifting-based MT scheme. Techniques for processing many-to-one mapping, non-referred pixels and sub-pel motion alignment are also discussed in detail. Section 3 describes the multi-layer MT structure. Correlation among motion vectors of neighboring blocks in the same layer and that among those of co-located blocks of different layers are also explored in this section. Section 4 presents the eight modes for representing different levels of correlation before proposing an R-D optimization algorithm for mode selection. Experimental results are given in section 5. Section 6 concludes the paper.

2. ADVANCED MOTION THREADING

The advanced MT technique assumes a lifting-based wavelet transform and it aims to reduce the number of artificially terminating/emerging threads in our original implementation of MT. The accuracy of motion alignment is increased to sub-pel level while maintaining perfect reconstruction.

2.1. Lifting-based motion-threading

The lifting structure is an efficient implementation of the wavelet transform with low memory and computational complexity [32, 33]. Every FIR wavelet filter can be factored into

a few lifting stages [33]. As an example, the lifting steps of a one-level biorthogonal 5/3 wavelet transform are shown in Figure 2. The input signal {x0, x1, …, x6} starts from the left hand side, and the wavelet coefficients {L0, …, L3} and {H0, …, H2} come out at the right hand side. From Figure 2, we have the lifting steps as:

$$\begin{cases} H_i = x_{2i+1} + a\,(x_{2i} + x_{2i+2}) \\ L_i = x_{2i} + b\,(H_{i-1} + H_i) \end{cases}, \qquad (1)$$

with a= -1/2, b=1/4.

This equivalently implements the convolution kernel of 5/3 wavelet transform (up to scaling).

In the lifting-based implementation, each lifting step only updates half of the nodes, and the original value of the updated nodes will not be needed in subsequent steps. The elementary lifting step is circled in Figure 2, which only involves three nodes. Thus the updated values can be saved in the same memory that holds the original values with in-place computation. For the wavelet transform, lowpass and highpass coefficients are interleaved. Each lifting step can be straightforwardly inversed with an inverse lifting unit. Thus the inverse transform structure is easy to obtain, as shown in the right side of Figure 2.

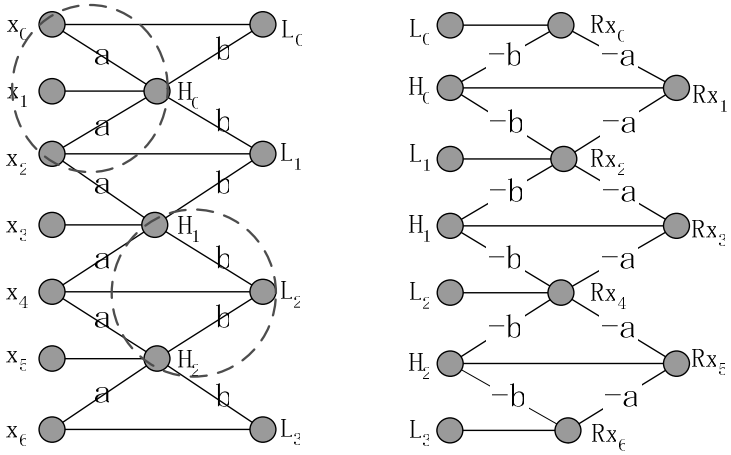


Figure 2: Lifting-based wavelet implementation of the 5/3 biorthogonal wavelet transform. Left: forward transform with lifting steps are shown in circles. Right: the inverse transform.

We implement the wavelet transform of motion threads using lifting. The video frames go through the lifting process step by step. We first update the odd frames to become highpass frames and then the even frames to become lowpass frames. Figure 3 shows the lifting-based 5-3 temporal wavelet structure, where each column is a frame, and each block represents a pixel. Motion estimation is done on the macroblock level and always performed from an odd frame to a pair of adjacent even frames. In each lifting step, either the pixels in odd frames or those in even frames are lifted along the motion threads.

The lifting steps can be formulated by:

$$\begin{cases} P_{H_i} = P_{F_{2i+1}} + a \times (MT(F_{2i}) + MT(F_{2i+2})) \\ P_{L_i} = P_{F_{2i}} + b \times (MT(H_{i-1}) + MT(H_i)) \end{cases}, \qquad (2)$$

where a=-1/2, b=1/4.

In each lifting step, either the pixels in odd frames or those in even frames are updated along the linked motion threads. $P_{F_{2i}}$ and $P_{F_{2i+1}}$ represent the current pixel to be updated in even and odd frames respectively; $MT(\cdot)$ represents the mapping pixels in the adjacent frames. P_{H_i} and P_{L_i} are the corresponding wavelet coefficients of the current pixel.

Since the length of each wavelet filter is short, as long as needed frames have entered the input buffer, related lifting steps can be executed without waiting for more frames to come in. Thus we implement a temporal extension of the line-based wavelet transform scheme used in JPEG2000 [2] with minimum buffer size during encoding and decoding (see [12]).

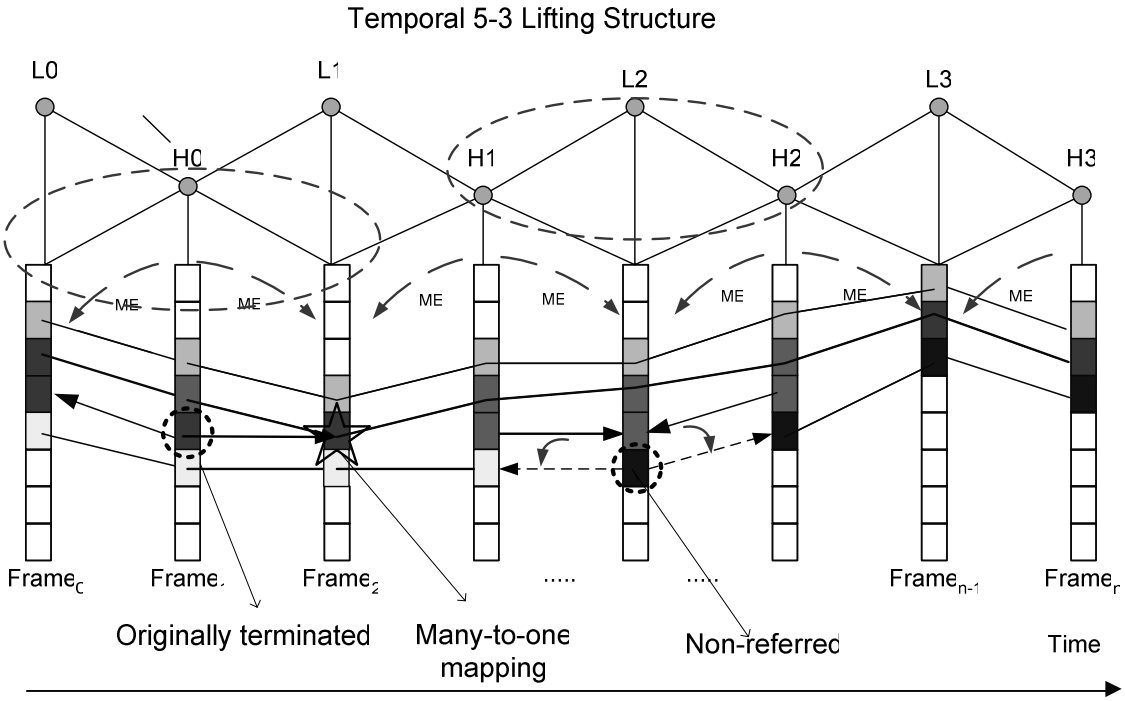


Figure 3: Lifting-based MT with bi-directional motion search.

2.2. Many-to-one mapping and non-referred pixels

Due to object movement, camera panning/zooming, and scene changes, there are usually different types of motion in real-world video. This would result in many-to-one mappings and non-referred pixels in MT. With lifting, we can carry on temporal filtering on pixels that are originally terminated in [11] due to many-to-one mappings. As shown in Figure 3, during each basic lifting step, the originally terminated pixel in Frame1 can be updated using both its left (forward) and right (backward) connected pixels instead of being stopped on the right. When the corresponding pixel in Frame2 is to be lifted, although several pixels in Frame1 are mapped to it, only the first mapped one of them is used. For a non-referred pixel in an even frame, which originally indicates the boundary of a motion thread, it will be linked on both sides using the motion vectors from adjacent motion threads. This way, each pixel is guaranteed to be linked in both forward and backward directions. This arrangement together with the lifting scheme ensures that the wavelet transform on motion threads is invertible.

Although we can always link each pixel in the current frame with a pair of pixels in the forward and the backward frames, sometimes this arrangement is counter-productive. For example, when camera panning, occlusion or scene change occurs, many pixels in the current frame might have good matches in the previous or next frame. Owing to the weak correlation among these pixels, linking them into motion threads will actually hurt the coding efficiency after the wavelet transform.

Since motion vectors are searched at the macroblock level in our implementation, mismatched pixels are also processed at the macroblock level. We mark a macroblock as a terminating one when it contains many terminating pixels. The terminating MB indicates the boundaries of the relative motion threads. The forward sum of absolute difference (SAD), backward SAD and the average SAD values of the MB are used in mode decision. Accordingly, besides the normal *Bid* mode in which a pixel is bi-directionally linked, two additional

MB modes, *Fwd* and *Bwd*, are used to represent the terminating status at the backward and forward directions, respectively.

For each MB, deciding the best mode among *Bid*, *Fwd* and *Bwd* is an optimization problem. As discussed in Section 1, the reconstruction error at the motion thread boundaries is larger than that within the threads, hence the boundary effects. To avoid these boundary effects due to artificial thread terminations, we favor *Bid* in mode decision. On the other hand, since terminating MBs only have uni-directional motion vectors, choosing the *Fwd* or *Bwd* mode saves the cost of motion vector coding. Details on mode selection are given in Section 4.

2.3. Threading with fractional-pixel accuracy

Lifting-based wavelet transform of motion threads enables sub-pel accuracy and perfect reconstruction in motion estimation. As indicated by the dashed curves in Figure 3, motion estimation is carried out from an odd frame to a neighboring even frame, either forwardly or backwardly. Without loss of generality, we consider the bi-directional case. We treat each elementary temporal lifting step as a bi-directional motion compensated prediction process that involves three consecutive frames. Figure 4 illustrates the first lifting step in which frame F_{2n+1} is lifted to a highpass frame. Black circles represent pixels at full-pel resolution, gray ones at ½-pel resolution, and white ones at ¼-pel resolution. Solid curves with solid arrows represent motion vectors generated from block motion estimation, and dashed curves with solid arrows represent motion vectors which are directly inverses of the solid curves. The dashed curve with a hollow arrow indicates that the pixel is not referred and the motion vector comes from the nearest dashed curve. As shown in Figure 4, in this lifting stage where frame F_{2n+1} is to be updated to a highpass frame, if pixel x_2 in frame F_{2n+1} refers to the half pixel between x_1 and x_2 in F_{2n}, then in the next lifting stage when F_{2n} is updated to a lowpass frame, x_2 in F_{2n} will accordingly refer to the half pixel between x_2 and x_3 in F_{2n+1}. In other words, the counterpart motion vectors are strictly kept in the opposite direction. Operations in the ¼-pel resolution follow the same procedure. Base on the elementary lifting operation, the reference frames in each lifting step can be reproduced at the decoder side, thus perfect reconstruction is guaranteed in wavelet reconstruction.

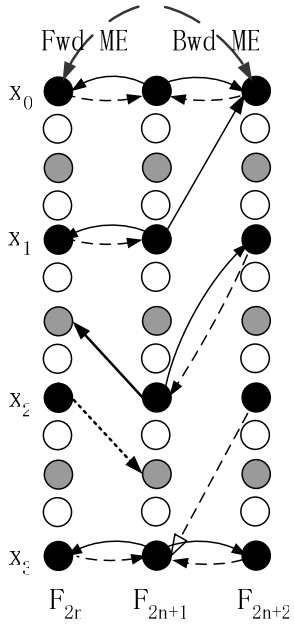


Figure 4: Elementary lifting operation at ¼-pel resolution.

3. MULTI-LAYER MOTION-THREADING

In our 3D wavelet video coder, temporal-domain wavelet transform is applied before the spatial-domain wavelet transform. A 4-level dyadic wavelet transform structure along the temporal axis (or motion threads) is shown in Figure 5. In our original MT implementation

[11], motion vectors are estimated and transmitted only at the highest layer. When performing the dyadic wavelet transform, we used different levels of aggregation of the motion vectors generated at the highest layer. However, by simply aggregating motion vectors generated from the highest layer, motion alignment in lower frame rate layers may be inaccurate. In addition, in the temporal scalability mode when the frame rate is less than the full rate, motion vectors at the highest layer have to be fully transmitted to generate motion vectors at lower layers. This incurs too much overhead in motion vector coding.

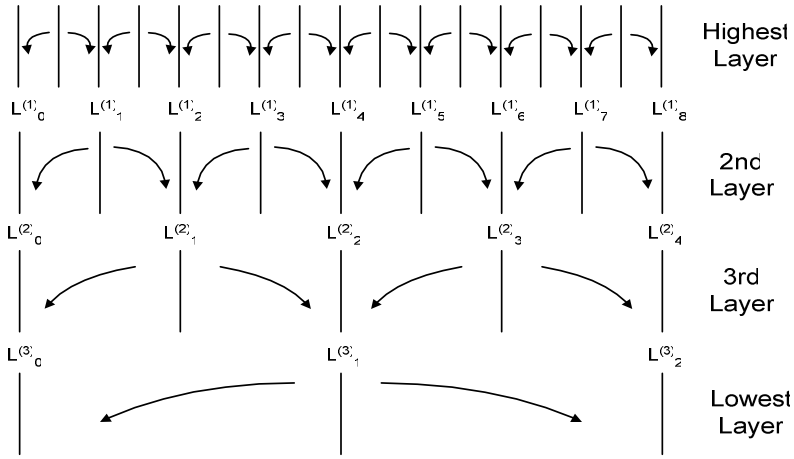


Figure 5: A 4-level dyadic wavelet transform structure along the temporal axis.

To rectify this shortcoming of [11], we propose in advanced MT to estimate and transmit motion vectors at each temporal decomposition layer [24]. Motion estimation is performed on the original frames (after appropriate downsampling) instead of the wavelet transformed lowpass frames. With multiple sets of motion vectors, it is imperative to exploit correlation among them for efficient coding. We first examine the correlation of motion vectors between two adjacent frames. The bi-directional motion estimations are always from odd frame to two neighboring even frames as shown in Figure 6. When the video frames undergo a constant speed motion, an assumption can be made that if a block at a position (x, y) in Frame_{2n+1} has a matching block at the position $(x + dx, y + dy)$ in Frame_{2n}, then using a uniform temporal motion extrapolation, the same block must move to position $(x - dx, y - dy)$ in Frame_{2n+2}. This assumption is also used in B-frame in the traditional video coding. When encoding the motion vectors, the forward and backward motion vectors have the same absolute value but with opposite signs, thus only one directional motion vector needs to be coded. This type of correlation is represented by two direct inverse (FwdDir/BwdDir) MB modes in motion estimation.

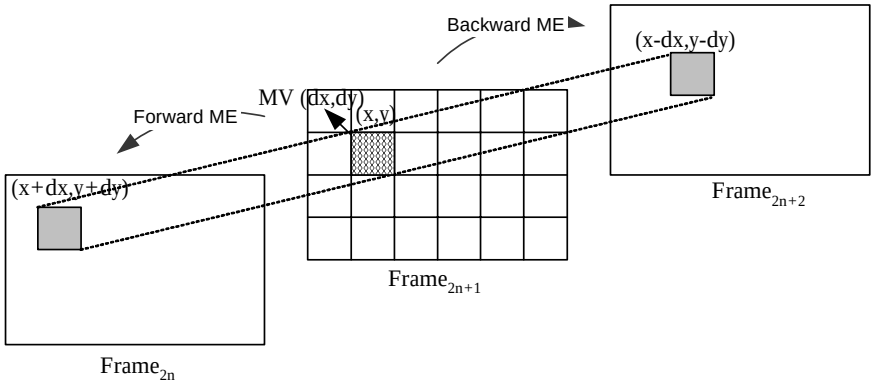


Figure 6: Motion correlation between two adjacent frames.

Next we discuss the inter-layer correlation based on the temporal multi-layer decomposition structure in Figure 5. Only the correlation between two neighboring layers is utilized. For each pair of neighboring layers, the block-based motion vectors at one layer are used for prediction of the motion vectors at the other layer.

There are two directions to utilize the inter-layer correlation. One is from high to low, which first estimates the motion vectors at the highest layer using traditional block motion estimation, and then predicts motion vectors of the lower layer from the higher layer. Similar to the temporal scalability problem we mentioned about the original MT, although this method ensures the motion accuracy, the higher layer motion vectors are always needed to be transmitted even at low frame-rate case. Hence the high-to-low method is not efficient for low bit-rate applications.

The other direction is low to high, which predicts from lower to higher layers. Since lower layer motion vectors do not depend on higher layer ones, the latter do not have to be transmitted in downsampled frame-rate applications. The disadvantage of this method lies in the precision of the motion vectors due to the long frame interval at lower level. For a 4-layer temporal decomposition, the interval is 8 frames. Furthermore, the prediction of the higher layer motion vectors is also affected.

To achieve nice frame-rate scalability, in this paper, we prefer the low-to-high method to predict the inter-layer correlation.

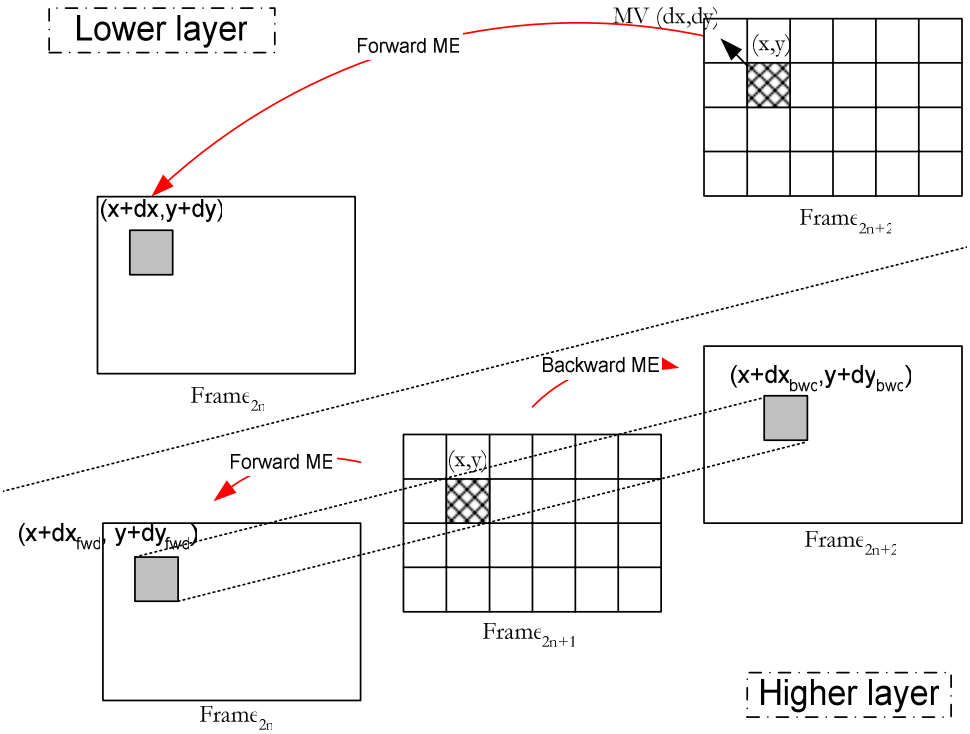


Figure 7: Correlation of motion vectors between two adjacent temporal layers.

Figure 7 illustrates the inter-layer correlation of motion vectors between two adjacent temporal layers. The inter-layer correlative modes assume that the sum of the absolute value of the forward and the backward motion vectors are equal to the motion vector of the same MB location in the previous layer. Therefore, one motion vector can be saved within each forward-backward motion vector pair. An extreme case of this assumption is that the forward and the backward motion vectors are equal, while no motion vector in the higher layer needs to be transmitted. Accordingly, three modes are designed to represent these correlation types.

4. CORRELATED MOTION ESTIMATION WITH R-D OPTIMIZATION

In the previous sections, we pointed out the mismatch problem and motion vector correlation in multi-layer motion threading. In this section, eight block modes are designed to represent different motion and correlation type. An R-D optimized correlated motion estimation (CME) [25] scheme is proposed to select from the designed modes based on a compound cost function. The cost function considers not only the matching correctness but also the motion vector bits.

4.1. Definition of the mode types

Figure 8 shows the flowchart of the proposed inter-layer motion estimation. Each MB in odd frame is first bi-directionally motion searched using the similar motion search technique in H.264 [31], here only the 16x16 MB mode is used. A pair of independent forward and backward MVs are then generated. After that, according to the previous analysis, we define eight MB modes to represent the diverse local motion properties. How to choose the modes is based on an R-D optimized criterion. The forward or backward motion vectors are updated according to the chosen mode. The next MB within the same frame may use the updated motion vectors to initiate the center value in the motion search. Then motion threads are assembled according to the chosen MB mode.

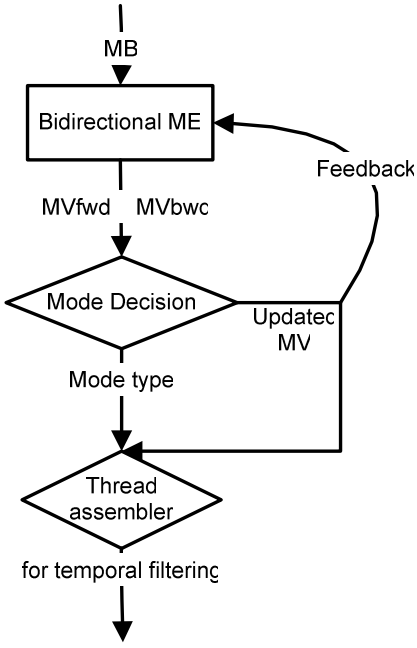


Figure 8: Flowchart of the correlated motion estimation.

As shown in Figure 9, each column denotes one frame, and each block can be considered as a macroblock. The current frame F_{2n+1} is divided into MBs. A dark gray MB represents an anchor block referred by the current MB. With camera zooming, the mapping block of the dark gray MB may not exist, as shown by the light gray blocks in the Fwd and Bwd mode in Figure 9. Each mode is defined as follows:

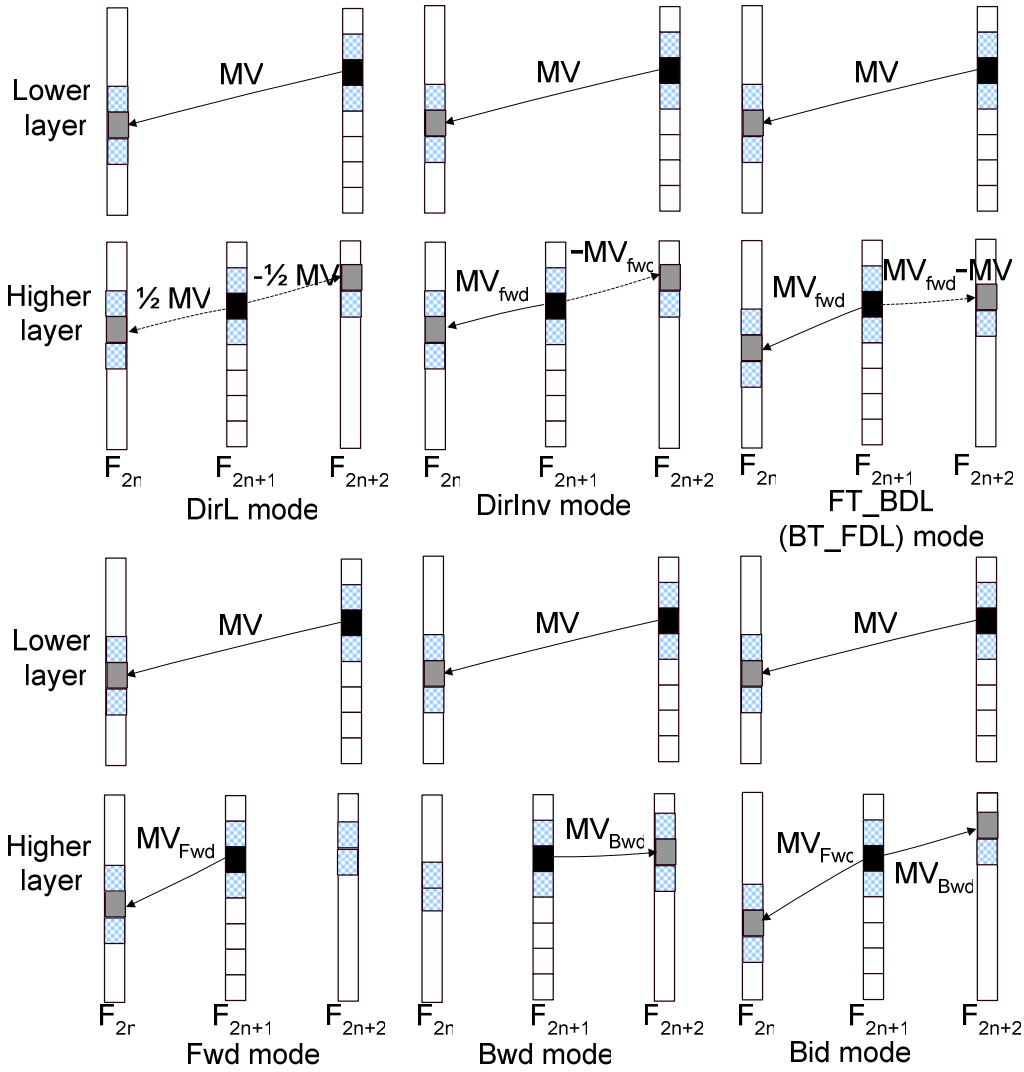


Figure 9: Eight motion modes in CME.

DirL: an inter-layer correlated bidirectional mode. The forward and the backward motion vectors use half the absolute value of the inter-layer predicting MV, with according inverse signs. No motion bits need to be transmitted in this mode. Some relatively smooth motion type, such as background motion, may belong to this mode.

FT_BDL (BT_FDL): two inter-layer correlated bidirectional modes. The forward (backward) MV is transmitted, while the relative backward (forward) MV is calculated as: $MVbwd = MVfwd \pm MVprevlayer$ ($MVfwd = MVbwd \mp MVprevlayer$). The selection of + or – operator depends on the backward or forward direction of the MVprevlayer. These two modes imply that the absolute values of the bi-directional motion vectors are unequal. Only one motion vector need to be sent. Between FT_BDL and BT_FDL modes, the one with less motion cost is chosen.

FwdDir (BwdDir): two intra-layer correlated bidirectional mode, where the forward and backward MVs have the same absolute value and inverse signs: MVbwd = – MVfwd (or MVfwd = – MVbwd). Only forward (backward) MV needs to be transmitted.

Fwd (Bwd): two single-directional modes. Only the forward (backward) MV is transmitted, and the other direction of the thread is terminated. When some particular motion such as occlusion occurs, truly-matched pixel may only exist either in the forward or in the backward frame. Therefore, the motion threads should be terminated in the mismatched direction.

Bid: the normal bidirectional macroblock mode. Both the forward and the backward MVs are transmitted.

4.2. R-D optimized mode decision

After a pair of independent forward and backward MVs are generated, from the designed eight MB modes, the one with the smallest cost is selected based on an R-D optimized criterion. A mode symbol is attached to identify the selected mode. The mode selection criterion is defined as: $Cost = \eta \cdot SAD + \lambda \cdot Bits_{motion}$, which consists two terms:

SAD is the sum of the absolute difference between the current macro-block and the motion compensated matching blocks. The SAD value evaluates the matching correctness of the forward and backward motion vectors. For the bi-directionally linked modes, such as Bid and Direct, the SAD term is the absolute difference between the original MB and the average of the forward and backward matching blocks. For the Fwd and Bwd modes, the SAD term takes only the one directional SAD value because the other direction is terminated. Pixels belonging to an MB with Fwd or Bwd modes are the boundaries of the relative motion threads. As mentioned in section 1, the wavelet synthesis error on the thread boundary is larger than that on the within-thread pixels. For the 5/3 filters, Fwd and Bwd amounts to P-frame coding in conventional standard video coding (e.g.,[31,34-36]), while other mode can be viewed as B-frame coding. H.264 [31] uses a factor of two for the motion cost in the R-D optimization to penalize B-frame coding (due to the additional motion cost). Here we have a somewhat opposite situation in 3-D wavelet video coding as we are biased against the Fwd and Bwd modes as they introduce artificial boundaries in MT. Thus, following the strategy adopted in H.264, we set η to two for the Fwd and Bwd modes and one for the other six modes.

$Bits_{motion}$ represents the bits for coding the motion vector difference (MVD). Since the mode distribution varies considerably for different sequence motion types and different temporal decomposition layers, the bits for identifying the mode symbol is accordingly Huffman coded after the motion estimated process. Thus the symbol bit cost is not counted in the $Bits_{motion}$ term during the motion estimation.

In Bid mode, both the forward and the backward motion vectors are coded, hence $Bits_{motion}$ consists of the bits for the bi-directional MVs; in the other modes, $Bits_{motion}$ only involves single-directional motion bits. λ is the weight parameter to control the R-D contribution of the motion cost in the cost function. Generally, at low bit-rate, the slope of the rate-distortion curve is sharper than that at high bit-rate, thus λ should be larger at low bit-rate to properly estimate the motion cost percentage. However, because of the non-recursive wavelet coding structure, motion estimation is performed prior to the transform and the quantization. Therefore, λ can only be set to a fixed value for all the bit rates in our coder (we note that a novel scheme based on scalable motion vector coding was recently proposed in [21] that effectively accommodates different λ*'s* in one coder).

In our experiments, we target at an average PSNR range of 30 to 40dB with the mean of 34dB. According to $PSNR = 10\log_{10}(255^2 / D)$, $D = Q^2/12$ under the high rate assumption for uniform scalar quantization, and the empirical formulae $\lambda = \sqrt{0.85} \cdot Q$ [36], we can relate λ to the target PSNR. For PSNR=34 dB, we have $\lambda \approx 16$ and we use λ = 16 in our experiments with six sequences.

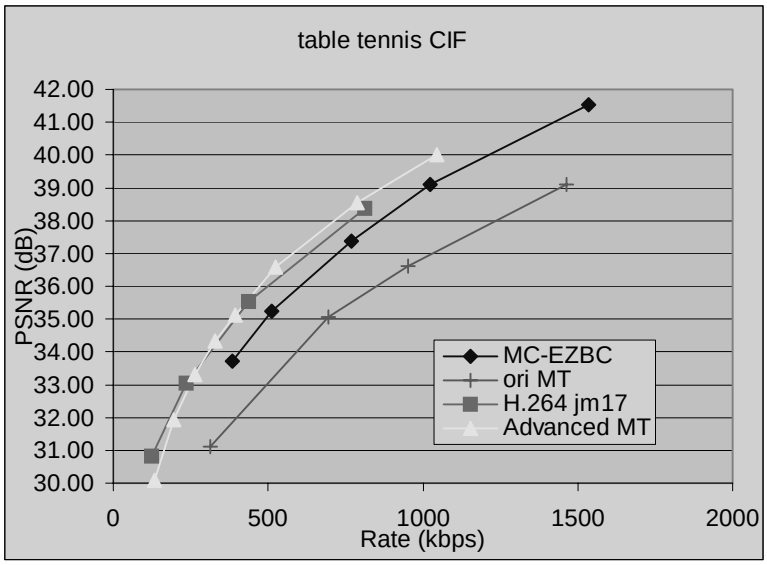
For each MB, the mode which has the smallest total cost value is selected. Various mode distributions are calculated according to different sequence and temporal layers. During the MB mode decision, the occurrence of each mode is counted for each temporal decomposition layer. The probability distribution of each mode is used in training the Huffman tables. Then the selected mode of each MB is Huffman encoded according to the trained tables. The Huffman tables for each temporal layer are written into the header of the transmitted bit-stream.

5. EXPERIMENTAL RESULTS

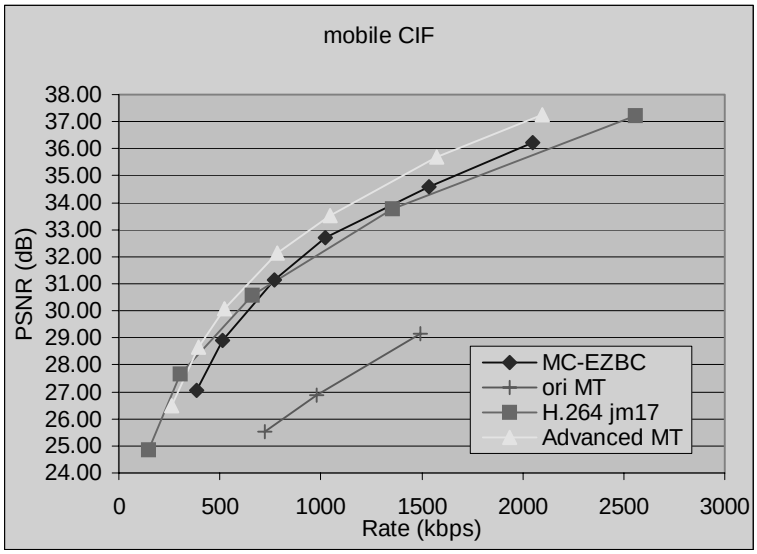
5.1. Coding performance comparison

We first compare the proposed advanced MT scheme with three benchmark coders: original MT [11], MC-EZBC [18] and H.264 [31] jm17. The result of MC-EZBC is cited from the result reported in the Excel file shown in [38]. In advanced MT presented here and the original MT coder [11], the input sequence is temporally filtered as a whole with the extension of the line-based implementation in JPEG2000 [2], without being explicit divided into GOPs [12]. Since the interval of the four-layer decomposed low pass frames is 16, the GOP size can be regarded as 16. The temporally filtered coefficient frames are further 3-layer spatially transformed with Spacl wavelet packet in JPEG2000 [2], where the three first-layer spatial high-bands are further vertically and horizontally analyzed, and then entropy coded as the operation in [14]. In advanced MT, ¼-pixel motion is used. Since the real motion range almost doubles when the temporal interval is doubled, the motion search-size is set as 32 for the highest temporal layer, and 64 for the second layer, and 128 for the rest layers. In H.264 coder, the GOP is set as a whole sequence with only one I frame, and three B frames are inserted into every two P frames. ¼ pixel with search-size 32 is applied to H.264. CABAC and R-D optimization are also turned on.

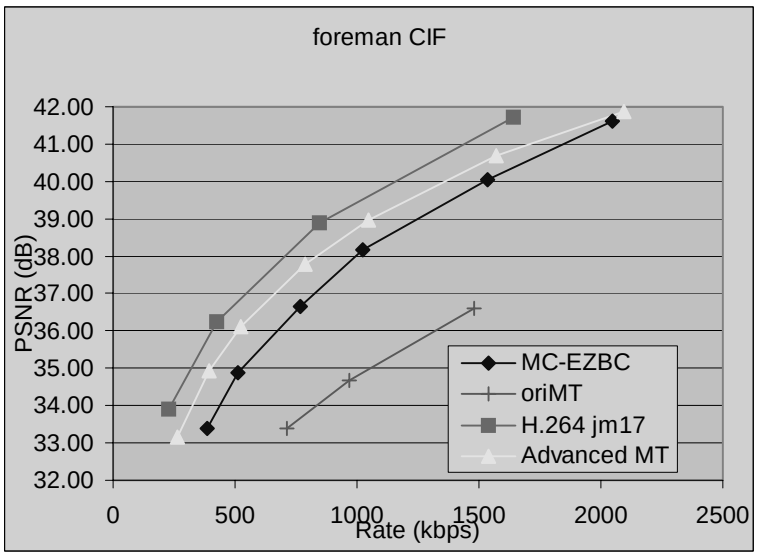
Experiments are performed on six MPEG standard CIF sequences. All the sequences are of 300 frames with the frame rate as 30Hz. Figure 10 shows that the original MT scheme achieves a performance similar to H.264 with Coastguard. However, with other sequences, the performance is not satisfactory. This is due to the boundary effect of large number of truncated motion threads and the insufficient integer motion accuracy. The proposed advanced MT improves the coding efficiency up to 6.0dB with Mobile and achieves a superior performance to H.264. The coding gain for Mobile mainly comes from the enhanced motion accuracy because Mobile possesses many texture details. With Foreman and Stefan, where the irregular motion generates a lot of truncated motion threads in the original MT, advanced MT improves about 3.5~4.0dB with the avoidance of truncation based on the temporal lifting structure. With these two sequences, advanced MT is still inferior to H.264 by 0.3~1.0dB due to the shortage in motion thread alignment for complex motion. However, we should note that the advanced MT can provide frame-rate and PSNR scalability which is not achieved by the single-layer H.264 bit-stream. With Coastguard, advanced MT improves about 1.6dB, and even outperforms H.264 1.0dB. The improvement of the advanced MT is considerably significant.



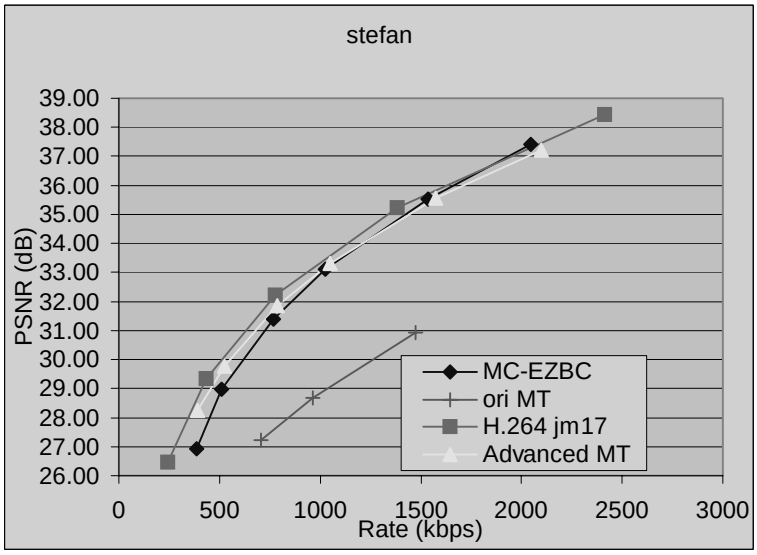
(a)


mobile CIF
PSNR (dB)
38.00
37.00
36.00
35.00
34.00
33.00
32.00
31.00
30.00
29.00
28.00
27.00
26.00
25.00
24.00
0
500
1000
1500
2000
2500
3000
Rate (kbps)
MC-EZBC
ori MT
H.264 jm17
Advanced MT

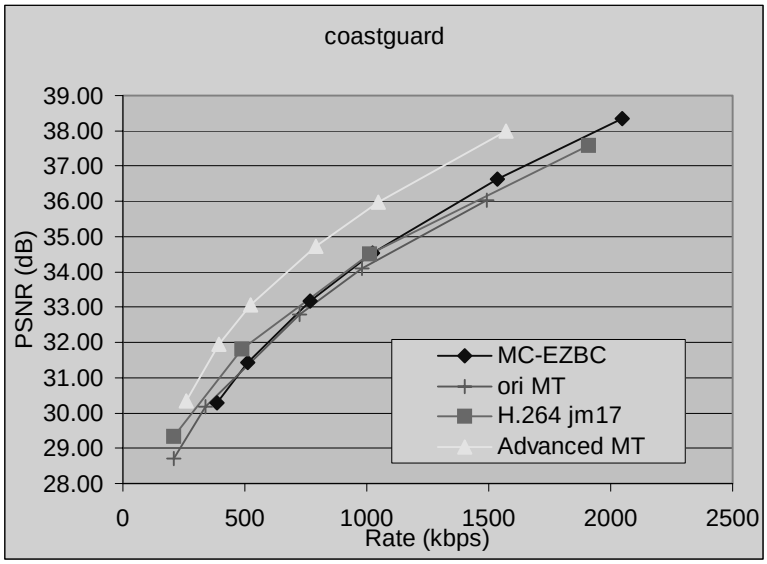
(b)


foreman CIF
PSNR (dB)
42.00
41.00
40.00
39.00
38.00
37.00
36.00
35.00
34.00
33.00
32.00
0
500
1000
1500
2000
2500
Rate (kbps)
MC-EZBC
oriMT
H.264 jm17
Advanced MT

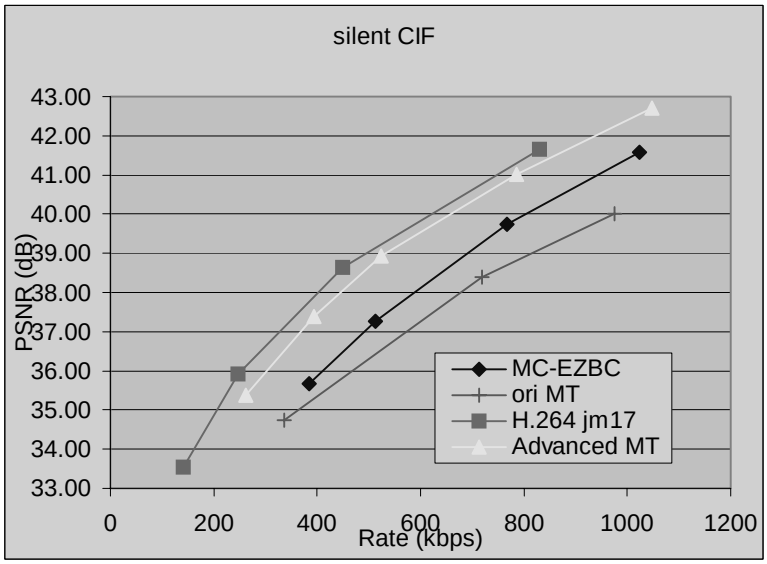
(c)


stefan
PSNR (dB)
39.00
38.00
37.00
36.00
35.00
34.00
33.00
32.00
31.00
30.00
29.00
28.00
27.00
26.00
0
500
1000
1500
2000
2500
3000
Rate (kbps)
MC-EZBC
ori MT
H.264 jm17
Advanced MT

(d)



(e)



(f)

Figure 10: Comparisons among MC-EZBC, H.264, original MT, and advanced MT.

5.2. Macroblock mode distribution

In this experiment, we investigate the performance of the mode-selective motion estimation. Three standard sequences: Foreman, coastguard and Mobile are used to represent diverse motion types. All of them are in CIF format, 30Hz, with 300 frames in all.

First we examine the distribution of the modes. The test sequences are temporally decomposed into four layers. For the lowest layer, DirL, FT_BDL and BT_FDL modes are disabled since the lower predictor is not available. Table 1 shows the distribution vs. decomposition layer of the three test sequences.

In the lowest layer, Bid mode takes a large percentage. As the layer gets higher, the distribution becomes biased towards the inter-layer and intra-layer modes. At the highest layer, DirL is about 50% of the total modes. With Foreman sequence, which contains more uneven and irregular motion such as face motion and panning, FT_BDL and BT_FDL modes are frequently used instead of DirL. In Mobile, the motion is more regular, thus the mode distribution concentrates around the DirL, FwdDir, and BwdDir modes where the forward and backward MVs are equal. The mode distribution of Coastguard is between Foreman and Mobile. The water wave, with which motion estimation fails to catch an accurate motion, leads to the relatively uniform mode distribution.

Table 1: Mode distribution percentage vs. temporal layer.

Mode Type	Lowest Layer4	Layer 3	Layer 2	Highest Layer 1
DirL	0	6.38	14.13	38.00
FT_BDL	0	34.72	36.14	23.85
BT_FDL	0	17.70	17.28	12.74
FwdDir	5.17	2.46	3.56	7.19
BwdDir	5.27	2.76	3.04	4.54
Fwd	3.44	2.06	1.21	0.41
Bwd	5.79	2.25	1.05	0.30
Bid	80.33	31.66	23.59	12.96

a) Foreman (%)

Mode Type	Lowest Layer4	Layer 3	Layer 2	Highest Layer1
DirL	0	33.95	60.37	67.94
FT_BDL	0	21.74	7.64	0.75
BT_FDL	0	8.20	3.36	0.55
FwdDir	12.70	8.87	14.85	24.57
BwdDir	5.54	4.44	6.30	5.42
Fwd	1.99	0.65	0.12	0.03
Bwd	0.58	0.28	0.10	0.02
Bid	79.19	21.87	7.25	0.72

b) Mobile (%)

Mode Type	Lowest Layer4	Layer 3	Layer 2	Highest Layer 1
DirL	0	9.59	24.78	57.63
FT_BDL	0	24.63	23.17	14.64
BT_FDL	0	11.19	9.93	6.08
FwdDir	14.49	13.13	13.14	9.73
BwdDir	15.08	11.55	8.50	4.44
Fwd	3.09	0.66	0.21	0.04
Bwd	1.39	0.97	0.25	0.03
Bid	65.95	28.28	20.00	7.40

c) Coastguard (%)

Next, we analyze the contribution of the layer-correlated ME Scheme. As shown in Table 2, IntraLayer denotes the scheme with only intra-layer modes such as FwdDir, BwdDir, Fwd, Bwd and Bid modes; CME is the proposed scheme which considers both the intra-layer modes and the inter-layer one such as DirL, FT_BDL and BT_FDL. The total bits consist of the motion bits and the mode bits.

With the same PSNR performance, the motion bits reduction is shown in Table 2. The bits reduction is 18.1% for Foreman, 19.2% for Coastguard, and 29.2% for Mobile. Since Mobile contains more regular motion, for most MBs, DirL mode is selected, thus the bits reduction is relatively larger. At low bit-rate (e.g., 384kb/s), the bit reduction is equivalent to about 0.3dB PSNR improvement.

Table 2: Motion bits reduction.

		Foreman	Mobile	Coastguard
Intra-Layer (kbps)	MV bits	96.5	49.4	69.2
	Mode bits	15.0	12.5	14.7
	Total bits	111.5	61.9	83.9
CME (kbps)	MV bits	66.8	25.2	43.2
	Mode bits	24.5	18,6	24.6
	Total bits	91.3	43.8	67.8
Total Reduction	(kbps)	20.2	18.1	16.1
	Per-centage	18.1%	29.2%	19.2%

Finally one reviewer asked for comparison of our results with that of [37] for the "mobile" sequence with the 5/3 filters. Our coder gives an average PSNR of 29.8, 33.3 and 37.0 dB at 500 kb/s, 1Mb/s and 2Mb/s, respectively. In contrast, results in [37] show average PSNR of 29.5, 33.1 and 36.5, respectively, at the same three bit rates.

6. CONCLUSION

In this paper, an advanced motion threading technique is proposed to improve the existing motion threading technique (MT) for wavelet video coding. With a lifting-based temporal wavelet structure, the boundary effect problem caused by artificial thread truncation in the original MT is well solved. Fractional-pixel motion can also be applied in the new structure. Multi-layer motion threading is used to achieve efficient frame-rate scalability. To efficiently reduce the motion cost as well as achieve good frame rate scalability, we analyzed both the interlayer and the intra-layer correlations. A novel R-D optimized correlated motion estimation (CME) scheme is proposed for the motion estimation process. Experimental results show the saving of the motion vector bits is up to 29.2% with Mobile sequence.

The lifting-based motion threading has many benefits. In the future work, adaptive-length wavelet filters can be applied into the lifting structure to adapt the various local properties along a video sequence.

ACKNOWLEDGEMENTS

The authors would like to thank Alexis Tourapis for his help in discussion about the direct mode design. Thank Ruiqin Xiong for his help on the encoder/decoder examination and partial experiments.

REFERENCES

[1] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*, Prentice Hall Englewood Cliffs, NJ 1995.

[2] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards, and Practice*, Kluwer Academic Publishers, 2001.

[3] G. Karlsson and M. Vetterli, "Three dimensional subband coding of video," *Proc. ICASSP'88*, New York, NY, 1988.

[4] J.-R. Ohm, "Three dimensional subband coding with motion compensation," *IEEE Trans. on Image Processing*, vol. 3, pp. 559-571, September 1994.

[5] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video", *IEEE Trans. Image Processing*, vol. 3, pp. 572-588, September 1994.

[6] C. Podilchuk, N. Jayant, and N. Farvardin, “Three-dimensional subband coding of video,” *IEEE Trans. on Image Processing*, vol. 4, pp. 125-139, February 1995.
[7] J. Tham, S. Ranganath, and A. Kassim, “Highly scalable wavelet-based video codec for very low bit-rate environment,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, pp: 369 –377, August 1998.
[8] A. Wang, Z. Xiong, P.A. Chou, and S. Mehrotra, “Three-dimensional wavelet coding of video with global motion compensation”, *Proc. DCC '99*, Snowbird, Utah, March 1999.
[9] S. Choi and J. Woods, “Motion-compensated 3-D subband coding of video,” *IEEE Trans. on Image Processing*, vol. 8, pp. 155-167, February 1999.
[10] B.-J. Kim, Z. Xiong, and W.A. Pearlman, “Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)”, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, pp: 1374 –1387, December 2000.
[11] J. Xu, Z. Xiong, S. Li, and Y.-Q. Zhang, “Three-dimensional embedded subband coding with optimized truncation (3D ESCOT)”, *Applied and Computational Harmonic Analysis*, pp. 290-315, 2001.
[12] J. Xu, Z. Xiong, S. Li, and Y.-Q. Zhang, ``Memory-constrained 3-D wavelet transform for video coding without boundary effects," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, pp. 812-818, September 2002.
[13] B. Pesquet-Popescu and V. Bottreau, “Three-dimensional lifting schemes for motion compensated video compression,” *Proc. ICASSP’01*, Salt Lake City, UT, May 2001.
[14] L. Luo, J. Li, S. Li, Z. Zhuang, and Y.-Q. Zhang, “Motion compensated lifting wavelet and its application in video coding,” *Proc. ICME’01*, Tokyo, Japan, August 2001.
[15] A. Secker and D. Taubman**,** “Highly scalable video compression using a lifting-based 3D wavelet transform with deformable mesh motion compensation,” *Proc. ICIP’02, Rochester, NY, September 2002.*
[16] A. Secker and D. Taubman**,** “Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression,” *IEEE Trans. on Image Processing*, vol. 12, December 2003.
[17] P. Chen and J. W. Woods, “Improved MC-EZBC with quarter-pixel motion vectors”, MPEG document, *ISO/IEC JTC1/SC29/WG11, MPEG2002/M8366*, Fairfax, VA, May 2002.
[18] P. Chen and J. Woods, “Bidirectional MC-EZBC with lifting implementation,” *IEEE Trans. on Circuit and Systems for Video Technology*, to appear.
[19] J.-R. Ohm, “Motion-compensated wavelet lifting filters with flexible adaptation,” *Proc. Intl. Workshop on Digital Communications*, Capri, 2002.
[20] D. Turaga, M. van der Schaar, and B. Pesquet-Popescu, “Reduced complexity spatio-temporal scalable motion compensated wavelet video encoding”, submitted to *IEEE Trans. on Circuit and Systems for Video Technology*, March 2003.
[21] A. Secker and D. Taubman**,** “Highly scalable video compression with scalable motion coding,” submitted to *IEEE Trans. on Image Processing*, May 2003.
[22] M. Flierl and B. Girod, "Video coding with motion-compensated lifted wavelet transforms," submitted to *Signal Processing: Image Communication*, June 2003.
[23] D. Turaga, M. van der Schaar, Y. Andreopoulos, A. Munteanu, and P. Schelkens, “Unconstrained motion compensated temporal filtering (UMCTF) for efficient and flexible interframe wavelet video coding”, submitted to *Signal Processing:Image communication*, June 2003.
[24] L. Luo, F. Wu, S. Li, and Z. Zhuang , “Advanced lifting-based motion-threading techniques for 3D wavelet video coding” , *Proc. VCIP’03*, Lugano, Switzerland, July 2003.
[25] L. Luo, F. Wu, S. Li, and Z. Zhuang , “Layer-correlated motion estimation and motion vector coding for 3D wavelet video coding”, *Proc. ICIP’03*, Barcelon, Spain, September 2003.

[26] Y.-Q. Zhang, S. Zafar, “Motion-compensated wavelet transform coding for color video compression”, *IEEE Trans. on Circuits and Systems for Video Technology*, vol.2, pp: 285 –296, September 1992.
[27] D. Marpe and H. Cycon, “Very low bit-rate video coding using wavelet-based techniques”, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, pp: 85 -94, February 1999
[28] E. Asbun, P. Salama, and E. Delp, “A rate-distortion approach to wavelet-based encoding of predictive error frames”, *Proc. ICIP’00*, Vancouver, Canada, September 2000.
[29] H. Park and H. Kim, “Motion estimation using low-band-shift method for wavelet-based moving picture coding”, *IEEE Trans. on Image Processing*, vol.9, pp.577-587, April 2000.
[30] Y. Andreopoulos, A. Munteanu , J. Barbarien, M. van der Schaar, J Cornelis, and P. Schelkens, “In-band motion compensated temporal filtering”, submitted to *Signal Processing: Image communication*, June 2003.
[31] “Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC,” in joint video team of ISO/IEC MPEG and ITU-T VCEG, JVT-G050, 2003.
[32] W. Sweldens, “The lifting scheme: A new philosophy in biorthogonal wavelet construction,” *Proc. SPIE Conf. Wavelet Applications in Signal and Image Processing*, 1995.
[33] I. Daubechies and W. Sweldens, “Factoring wavelet transforms into lifting steps”, *J. Fourier Anal. Appl.*, vol. 4, pp. 247-269.
[34] M. Flierl and B. Girod, “Generalized B pictures and the Draft H.264/AVC video-compression standard,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, pp: 587 -597, July 2003.
[35] G. Sullivan and T. Wiegand, “Rate-distortion optimization for video compression,” *IEEE Signal Processing Magazine*, vol. 15, pp. 74-90, November 1998.
[36] T. Wiegand and B. Girod, “Lagrange multiplier selection in hybrid video coder control,” *Proc. ICIP’01*, pp. 542-545, Thessaloniki, Greece, October 2001.
[37] D. Taubman, “Successive refinement of video: fundamental issues, past efforts and new directions,” Proc. VCIP’03, 2003.
[38] P. Chen and J. W. Woods ,”Contributions to Interframe Wavelet and Scalable Video Coding”, ISO/IEC JTC1/SC29/WG11 MPEG2002/M9034 October 2002, Shanghai, China