LAYERED MOTION ESTIMATION AND CODING FOR FULLY SCALABLE 3D WAVELET VIDEO CODING

*Ruiqin Xiong**[1], *Jizheng Xu*[2], *Feng Wu*[2], *Shipeng Li*[2], *Ya-Qin Zhang*[2]
[1] Institute of Computing Technology, Chinese Academy of Sciences, 100080, Beijing
[2] Microsoft Research Asia, 100080, Beijing

ABSTRACT

This paper proposes a framework of scalable motion estimation and coding with the structure of multi-layers for 3D wavelet video coding. The motion representation consists of multiple layers. The encoder uses motion of all layers to perform analysis, while the decoder may receive only part of motion for synthesis. Different from other schemes, each layer of motion is a point optimized at a certain range of bit-rate. We observe that the distortion introduced by motion mismatch is highly independent with the rate for texture in a wide range. Therefore, to make the best trade-off between motion and texture under the constraint of a given bit rate, a motion layer decision algorithm is used to find the appropriate number of motion layers to be included into the bit-stream. The proposed framework also supports the spatial and temporal scalabilities of motion. Experimental results show significant improvement at low bit-rates and nearly no loss at high bit-rates with layered motion coding and optimal motion decision. The performance is approaching to the convex hull of those with multiple sets of non-scalable motion.

1. INTRODUCTION

In general, texture bitstreams produced by most of wavelet video coders are fully scalable by using bit-plane coding. The coefficients are reordered according to their relative significances to achieve the best performance under the constraint of a given bit rate. Furthermore, the spatial and temporal scalabilities are inherent because of the multi-resolution representation of wavelet. However, these features are not sufficient for a fully scalable wavelet video coder to be efficient over a broad range of bit-rate. In most existing wavelet video coders, motion is non-scalable and can only be coded losslessly as a whole. The indivisible motion can not favor both high bit-rates and low bit-rates simultaneously. To achieve a rate-distortion optimized trade-off between motion and texture at an arbitrary bit-rate, a scalable representation of motion is desirable.

Scalable coding of motion generates an embedded bitstream consists of multiple layers, which represents a successively refinement of motion. The encoder uses the finest motion constructed from all motion layers to perform analysis. However, when the bit rate is very limited, the decoder may receive only part of the motion layers for synthesis, giving a higher priority to textures. There are two fundamental problems to be solved in this situation. The first one is how to optimize every truncation of motion. Another one is how to find the best rate allocation between motion and texture.

An optimal scalable representation of motion should try to maximize the motion accuracy when it is truncated at a given motion rate. The motion accuracy can be measured by the motion compensated prediction (MCP) error, as in equation (1), where F_{anchor} and F_{refi} are the video frames involved in MCP and M is the motion field. Huang *et al.* [1] proposed a simple two-layer motion coding for MC-EZBC [2] by roughly dividing the quad-tree to two even parts. It does not try to find the optimal motion for the truncated motion rate. Taubman [3] introduced a novel idea of adopting the same wavelet-transform and fractional bit-plane coding techniques as those used in JPEG2000. A linear model is assumed in [3] to infer the impact of motion error on video distortion and to design the transmission order of motion information. Since the split of motion are not done in the motion estimation, it may be difficult to make the split points to be optimal.

$$\min_M \{ \| F_{anchor} - MC(F_{ref1}, F_{ref2}, ..., F_{refn}, M) \| \} \quad (1)$$
$$given \ \mathrm{R}_\mathrm{M} \le \mathrm{R}$$

In this paper, we propose a framework of layered motion estimation and coding. In the proposed layered scheme, the rate-distortion constrained motion estimation is applied to each layer, that is, each layer of motion is optimal within a certain range of bit-rate.

When the decoder uses only part of motion for synthesis, motion mismatch must exist between the encoder and the decoder, which resulted in extra distortion in decoded video. We have observed that the distortion introduced by motion mismatch is almost completely independent of the rate for texture in a wide range. Based on this observation, we propose an optimal motion layer decision algorithm to select the best number of motion layers to be transmitted at any given bit rate. Furthermore, our framework can support both the temporal and spatial scalabilities for 3D wavelet video coding.

The paper is organized as follows. Section 2 describes the framework of layered motion coding. A scheme for optimal motion layer decision is also introduced in this section. Section3 discuss the detailed implementation of the scheme. Experimental results are given in Section 4 to show the performance. Section 5 concludes this paper.

2. THE FRAMEWORK OF LAYERED MOTION CODING

2.1. Layered Motion Coding

[1] This work has been done while the author is with Microsoft Research Asia.

In the non-scalable video coding, rate-constrained motion estimation is usually used to make the best trade-off between motion and texture by minimizing

$$J = D_{MCP} + \lambda \cdot R_{motion} \ . \qquad (2)$$

where D_{MCP} can be SAD or SSD and λ is the Lagrangian factor to adjust the trade off, which should have a large value for low rates but have a small value for high rates. In a scalable coder, the decoding rate is unfixed and the motion scalability is desirable. In general, we can present the motion of each macroblock with quad-trees and code each of them in a scalable way independently to obtain a fully scalable motion data. However, the spatial and temporal correlations among motion vectors are critical for efficiently motion coding. It is necessary to exploit the dependencies among motion vectors of every macroblock. Therefore, this paper proposes a layered motion coding, which not only provides the motion scalability at a certain extend but also maintains the high motion coding efficiency.

Figure 1 illustrates the framework of the proposed layered motion coding. We generate an embedded bitstream for motion, which consists of one base layer and a few enhancement layers. A coarse motion field can be reconstructed from the base layer of motion bitstream and it can be successively refined by decoding the subsequent enhancement layers. The base layer of motion corresponds to a low bit-rate and is generated using a relatively large λ, while the enhancement layers of motion correspond to higher bit-rates and are generated using a set of relatively small λ.

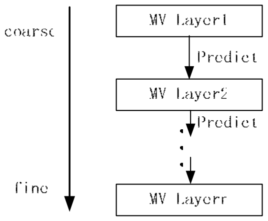


Figure 1: The proposed layered motion coding

Both the base layer and the enhancement layers can be obtained by any existing motion estimation and coding techniques, with a few small modifications. For fast and efficient estimation and coding of the enhancement layers, we should fully take advantage of the motion vectors (MVs) and block partition modes already coded in its previous layer. The coded MVs of a block in the previous layer can be used as one candidate of predicted MV and the block partition modes obtained from the previous layer can be regarded as the initial status of partitioning for the co-located block in the current layer.

The above arrangements of using different λ at different layers enable rate scalability of motion. In addition, the temporal and spatial scalability of motion can be supported as well. In 3D wavelet video coding, a natural way to support the temporal scalability is to bind the motion vectors with the temporal high-band coefficients at the same level and drop them together if it is necessary.

To support different spatial resolutions, the motion layers can be produced at different resolutions. For example, we can perform motion estimation on QCIF format of a video in Layer 1 and perform motion estimation on CIF format of the video in Layer 2. The motion obtained in Layer 1 will be up-sampled into a CIF resolution before it is used by Layer 2 as prediction.

2.2. Motion Mismatch and Motion Decision Scheme

For high coding efficiency, it is critical to fully exploit temporal correlation among frames with motion. In scalable video coding with layered motion, a highly accurate motion field is required for encoding and decoding at high bit-rates. However, for low bit-rates, only a subset of motion layers for a coarse motion field should be transmitted to make sure enough bits are allocated to texture information.

While decoding a part of motion bitstream, the video quality can not be fully recovered because of the motion mismatch even if all of the texture coefficients are decoded losslessly. The mismatch in motion results in mismatch in motion compensated prediction images:

$$Mismatch = \left\| MC(F, M_{Encoder}) - MC(F, M_{Decoder}) \right\| \qquad (3)$$

where $\mathrm{M_{Encoder}}$ is the motion at the encoder and $\mathrm{M_{Decoder}}$ is the motion at the decoder. F is the set of video frames involved in MCP. With many well-designed experiments, we have found that the video distortion resulted from motion mismatch is nearly constant in terms of MSE at a wide range of bit-rate for texture. This means the distortion from motion mismatch is highly independent with that resulted from texture quantization. This fact facilitate the motion layers selection in the final bitstream for a given bit rate.

When bit-rate decreases, we have to switch from a fine version of motion to a coarse one. The problem is how we can decide the proper bit-rate for motion layers' switching. In other words, we should select the most appropriate number of motion layers to be transmitted when the total bit-rate is given. Here we propose a motion layer decision method as follows:

Let's use $D_T(R_T)$ to denote the rate-distortion function of video decoded with lossless motion, where R_T is the rate for truncated texture bitstream. Actually $D_T(R_T)$ represents the R-D property of texture and can be estimated from the R-D information of coefficients in each subband. Let's denote the motion layers as M_1, M_2, …, M_n and the motion rates as R_{M1}, R_{M2}, …, R_{Mn}, respectively. $M = M_n$ is the finest motion. The distortion introduced from motion mismatch is denoted as

$$D_{Mismatch}(M, M_i) \ \ (1 \le i \le n)$$

and can be calculated according to the equation (3). Obviously, we have $D_{Mismatch}(M, M) = 0$. We can also obtain the mismatch of each motion version just by losslessly decoding all the texture coefficients with that motion. It is practical since the number of motion layers is few. Obtained the D_T and $D_{Mismatch}$, we can estimate the distortion $D_i(R)$ of video decoded using motion M_i:

$$D_i(R) = D_T(R - R_{Mi}) + D_{Mismatch}(M, M_i) \qquad (4)$$

Therefore, for given bit-rate R, select the motion M_i, where

$$i = \arg\min_i \{ D_T(R - R_{Mi}) + D_{Mismatch}(M, M_i) \} \qquad (5)$$

3. IMPLEMENTATION IN OUR CODER

In the framework of advanced motion threading (MTh) 3D wavelet coding [4][5], multi-level motion compensated temporal wavelet transforms are applied to the input video sequence before spatial transform. Motion at each level of temporal transform is estimated and coded independently. The motion at each level consists of a few layers optimized for a series of rate ranges with different λ.

In the base layer, motion estimation and coding are performed in the same way as in conventional coder. Pictures are divided into macroblocks with size of 16x16 and each of them can select a partition mode from 16x16, 8x16, 16x8, 8x8 and 8x8+ as shown in Figure 2. The 8x8+ mode means each 8x8 subblock in the macroblock can be further split into 8x4, 4x8 or 4x4 mode. Each subblock is assigned a motion vector which is searched by rate-constrained motion estimation. Rate-distortion optimization of motion is achieved by minimizing equation (2).

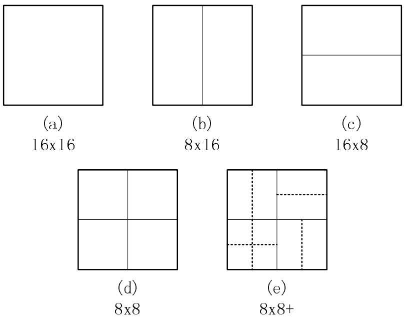


Figure 2: The partition modes of macroblock.

For the enhancement layer, initial motion vectors and block partition status are obtained from the previous layer before it starts its own motion estimation. If the adjacent two layers have different resolution, a conversion is needed as shown in Figure 3, where a macroblock in a low-resolution layer corresponds to four macroblocks in the current high-resolution layer. If they have the same resolution, the partition mode in previous layer is directly set as the starting point of further partition in this layer.

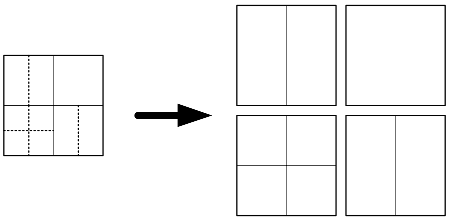


Figure 3: Conversion of partition mode. A macroblock with 8x8+ mode in previous layer/resolution corresponds to four macroblocks in current layer/resolution, with 8x16, 16x16, 8x8, 8x16 mode, respectively.

Since the initial status of a macroblock can be one of the modes, a, b, c and d, as shown in Figure 2, the refinement routes from the initial status to the final macroblock partition mode are given as follows.

$$a \rightarrow \{a,b,c,d,e\}$$
$$b \rightarrow \{b,d,e\}$$
$$c \rightarrow \{c,d,e\}$$
$$d \rightarrow \{d,e\}$$

Furthermore, the partition mode can be efficiently coded using the initial status as context.

In addition, the initial motion vectors obtained from previous layer are used to improve the accuracy of predicted motion vectors. If Ma, Mb and Mc are neighboring motion vectors of a block as shown in Figure 4 and Mp is its initial motion vector, the motion prediction is calculated according to equation (6b) instead of equation (6a) used in the base layer. Mp is used as several identical candidates. The number of occurrences of Mp can be adjusted according to the reliability of Mp. It can be measured by the cross-layer size difference of the blocks which own the motion vector Mp. A special case is to use Mp directly as the predicted motion vector and it can work well for most cases.

$$MVp = \begin{cases} median(M_a, M_b, M_c) & (6a) \\ median(M_a, M_b, M_c, M_p, ..., M_p,) & (6b) \end{cases}$$

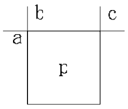


Figure 4: The prediction of motion vector.

With the usage of the initial motion vector and partition mode as a context, the CME modes in [5][6] can be used in the same way to efficiently code the motion which is well predicted from previous layer. These modes can greatly reduce the overhead of layered coding of motion.

4. EXPERIMENTAL RESULTS

Extensive experiments have been carried out on several standard test sequences to show the performance of the proposed scheme.

Firstly, we studied the distortion caused by motion mismatch. A three-layer motion is produced for *Foreman* CIF sequence with 300 frames. M_1, M_2, M_3=M are the three motion layers. Figure 5 shows the texture-rate-distortion curves of video decoded using the three different motion layers. The R-D curves estimated according to equation (4) for M_1 and M_2 are also shown. We can see the estimation is very accurate. It verifies the independence of distortion caused by motion mismatch with that from texture quantization.

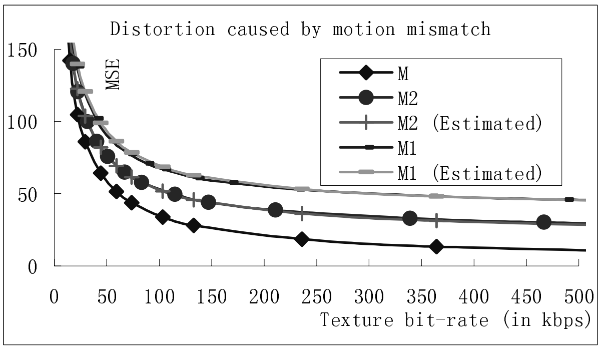


Figure 5: Distortion from motion mismatch.

Then we tested the performance of layered motion on *Foreman* and *Bus sequences* in CIF format. A four-level temporal transform is adopted and the motion at each level consists of two layers, the first one for QCIF with λ =64 and the other for CIF with λ =8 or 16. Figure 6 and Figure 7 show the performance of layered motion coding with optimal motion layer selection for *Bus* and *Foreman sequences*. They are also compared with results of non-scalable motion scheme. Figure 6 and figure 7 demonstrate that the proposed layered motion coding can always catch the nearly best performance compared with those two coding curves with single motion layer, both at the low end and at the high bit rate end.

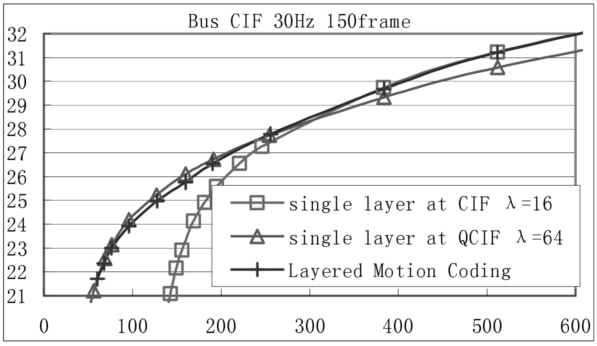


Figure 6: Performance of layered motion for Bus.

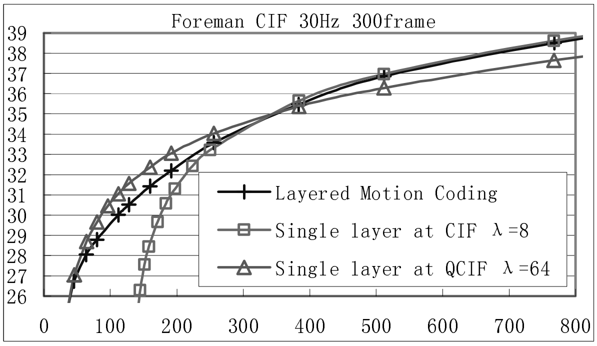


Figure 7: Performance of layered motion for Foreman.

Figure 8 presents the spatial scalability results generated by our scheme, using the same parameters as in the figure 7. We assume a total bit rate of 96kbps. The visual quality of the QCIF video in the figure is much better than the one in CIF format. So there exists a trade off between the SNR scalability and spatial scalability. So we also demonstrate that it is necessary to provide fully scalable feature to satisfy various requirements, which is enabled in our framework.

5. CONCLUSIONS

A framework of scalable motion coding with layered structure is proposed in this paper. Each layer of motion is optimized at a certain range of bit-rate. The distortion caused by motion mismatch is independent with that from texture quantization. Therefore we can calculate the distortion resulted from motion mismatch and estimate the overall performance when motion mismatch exists. A decision scheme is used to find the appropriate number of motion layers transmitted at a given bit-rate. The overall performance is approaching to the convex hull of those with multiple sets of non-scalable motion.

6. REFERENCES

[1] H. M. Hang, S. S. Tsai, T. Chiang, “Motion Information Scalability for MC-EZBC: Response to Call for Evidence on Scalable Video Coding”, ISO/IEC JTC1/SC29/WG11/m9756, July 2003, Trondheim.

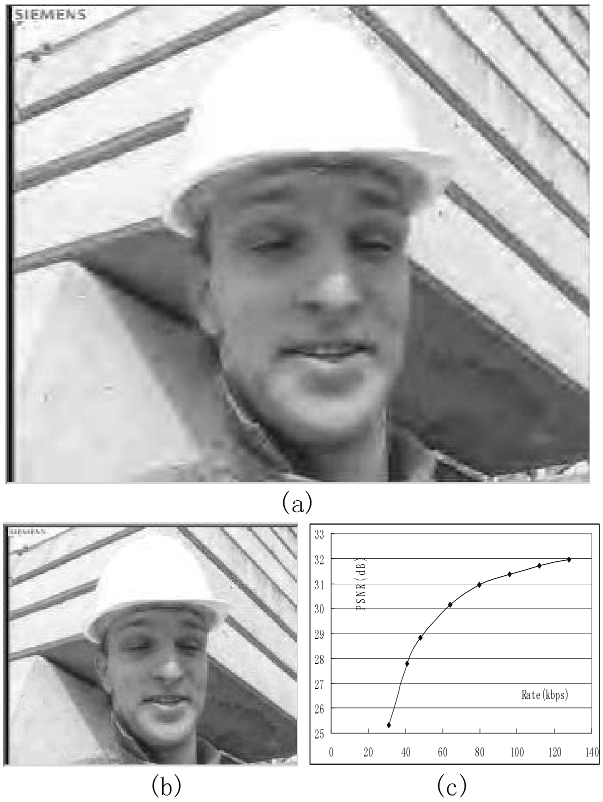
[2] P. Chen, J. W. Woods, “Improved MC-EZBC with quarter pixel motion vectors”, JVT proposal, ISO/IEC JTC1/SC29/ WG11, MPEG2002/M8366, Fairfax, VA, May 2002.

[3] D. Taubman, A. Secker, “Highly Scalable Video Compression with Scalable Motion Coding”, *Proc. ICIP* 2003,

[4] 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 vol.*10, pp.290-315, 2001.

[5] L. Luo, F. Wu, S. Li, Z. Zhuang, “Advanced lifting-based motion threading technique for 3D wavelet video coding,” *Proc of SPIE VCIP2003*, vol.5150, pp.707-718, Jul.2003.

[6] R. Xiong, F. Wu, S. Li, Z. Xiong, Y.Q. Zhang, “Exploiting temporal correlation with block-size adaptive motion alignment for 3D wavelet coding,” *Proc. of SPIE VCIP2004*, San Jose, California, USA, Jan.2004.



(a)

(b)

(c)

Figure 8: Demonstration of the spatial scalability results.
(a) CIF video at 96kbps (Average 29.2dB, Frame0 at 32.5dB)
(b) QCIF video at 96kbps (Average 31.4dB, Frame0 at 36.3dB)
(c) The PSNR versus bit-rate curve of decoded QCIF video.