

A Wavelet Interlaced Video Coding Framework

L. M. Mei, *Member, IEEE*, H. R. Wu, and D. M. Tan

Abstract-- A Discrete Wavelet Transform (DWT) based codec with EBCOT/ adaptive coding mode switching/ low-band-shift (LBS) is presented in this paper. The adaptive coding mode switching is used for both intra- and inter-frame coding, respectively, to improve the rate vs. distortion (R-D) performance. EBCOT, the coding engine used in JPEG2000 is adopted in this framework because of its embedded R-D optimization capability which is most suitable for the application of a perceptual distortion measure (PDM). LBS is used for motion estimation (ME) in DWT domain, while context based MQ coding is adopted to code the motion vector (MV). The context design of MV coding is derived from H.264.

I. INTRODUCTION

The DWT based image/video coding has been an extensive research topic for the past decade and a half [1]. Although current international and industry video compression standards all adopt the Discrete Cosine Transform (DCT) based techniques, DWT based coding has remained a topic of interest in literature due to structural distortions (e.g. blocking) in DCT coders and the scalability issue [2]. Most existing DWT based video coders operate at low to very low bit rates and are intended for videoconferencing and mobile telephony. These applications generally work under the frame mode. However, there are other applications, such as digital TV, where pictures are presented in interlaced (field) mode. The odd and even fields in interlaced pictures are out of phase. Consequently, when both fields are arranged in a single frame, phase mismatch appears on vertical edges between the fields. Therefore, applying frame mode coders to interlaced pictures is inefficient, since interlacing artifacts (due to phase mismatch) translate to bitrate intensive high frequency information in the transform domain. A DWT video codec that addresses this problem has been presented in [2]. Here, the focus is on the improvement of the Motion Compensation (MC) and the Motion Vector (MV) coding.

II. THE PROPOSED CODER

Since improved R-D performance is the main goal here, the traditional MC-2D interlaced approach is adopted here as illustrated in previous reports [2]. The DWT is performed on the input frame or individual field depending on whether it is in frame or field mode. The choice of coding mode is due to the following fact: although the field mode may improve the compression efficiency of the data, it may also impede the overall compression due to the header (overhead) of the second picture field. The allocation of bits is a tradeoff between higher compression ratio of the data and the

overhead. The intra-coder basically uses the JPEG2000 Core coding engine with minor modification [3]. For the inter-frame pictures, the MC is performed in the transform domain, and the residue produced by the differential pulse code modulation (DPCM) along the temporal direction is sent to the inter-frame embedded blocking coding with optimized truncation (EBCOT) coding engine. The DWT based video coders eliminate blocking artifacts that are commonly associated with block based DCT video coders, especially at low bit rates. For more details of wavelet block (WB) the readers are directed to [2]. ME and motion prediction (MP) can then be performed in the DWT domain based on block matching algorithm. LBS [4] is used to generate the reference WB in order to overcome the shift-variant problem. A tree-based implementation of LBS is developed because of its fast execution speed. Here, the notation of data structures in node and tree is used to simplify explanation of LBS. A node can be described by the following C style pseudo code:

```
nodeLBS {
  int shiftHorizontal;      // horizontal shift
  int shiftVertical;       // vertical shift
  double ** data;         // data before DWT
  int level;              //current DWT level
  int weight;            // weight associated with the current node
  // four children. They are UNSHIFTED,
  // SHIFT_HORIZONTAL, SHIFT_VERTICAL,
  // SHIFT_DIAGONAL, respectively.
  nodeLBS * children[4];
};
```

The LBS procedure will generate a quadric tree, i.e. each node in the tree will have four children except for the leaves. The root node of the tree contains the input image data. Its four direct children contains the un-shifted, horizontally shifted by 1 pixel, vertically shifted by 1 pixel, and diagonally shifted (shifted on horizontal and vertical direction by 1 pixel, respectively), version of data, respectively. For each one amongst these four children, the data is decomposed into LL, HL, LH, and HH bands, respectively. The next level of LBS will be carried on based on the coefficients of the LL bands. For an n-level LBS-DWT system, there are $(4^{n+1} - 1) / 3$ nodes including the root. The size of data in each individual nodeLBS depends on the level. The width and height of the input data is designated as *widthInput* and *heightInput*, respectively. Then the width and height of the node at level n is denoted as $(widthInput) / 2^{n-1}$ and $(heightInput) / 2^{n-1}$, respectively. If the ME is organized to compare the current

WB with the reference WB, and the measure for ME is the Mean Absolute Difference (MAD), then the computation of MAD is shown as in Equ. (1) where (x_i, y_i) is the initial point of the

$$\begin{aligned}
MAD_k(dx, dy) = & \sum_{i=1}^3 \sum_{x_i=x_{i,k}}^{x_{i,k}+M/2^i-1} \sum_{y_i=y_{i,k}}^{y_{i,k}+N/2^i-1} \left\{ \left| HL_{cur}^{(i)}(x_i, y_i) \right. \right. \\
& \left. \left. - HL_{ref}^{(i)}\left(dx\%2^i, dy\%2^i, x_i + \left\lfloor \frac{dx}{2^i} \right\rfloor, y_i + \left\lfloor \frac{dy}{2^i} \right\rfloor\right) \right| \right. \\
& + \left| LH_{cur}^{(i)}(x_i, y_i) - LH_{ref}^{(i)}\left(dx\%2^i, dy\%2^i, x_i + \left\lfloor \frac{dx}{2^i} \right\rfloor, \right. \right. \\
& \left. \left. y_i + \left\lfloor \frac{dy}{2^i} \right\rfloor\right) \right| + \left| HH_{cur}^{(i)}(x_i, y_i) - HH_{ref}^{(i)}\left(dx\%2^i, dy\%2^i, \right. \right. \\
& \left. \left. x_i + \left\lfloor \frac{dx}{2^i} \right\rfloor, y_i + \left\lfloor \frac{dy}{2^i} \right\rfloor\right) \right| + \sum_{x_3=x_{3,k}}^{x_{3,k}+M/2^3-1} \sum_{y_3=y_{3,k}}^{y_{3,k}+N/2^3-1} \left\{ \left| LL_{cur}^{(3)}(x_3, \right. \right. \\
& \left. \left. y_3) - LL_{ref}^{(3)}\left(dx\%2^3, dy\%2^3, x_3 + \left\lfloor \frac{dx}{2^3} \right\rfloor, y_3 + \left\lfloor \frac{dy}{2^3} \right\rfloor\right) \right| \right\} \quad (1)
\end{aligned}$$

i th-level subbands in the k th wavelet block. $HL_{ref}^{(i)}$, $LH_{ref}^{(i)}$ and $HH_{ref}^{(i)}$ are the i th-level HL, LH and HH subbands of the reference frame, respectively. $LL_{ref}^{(3)}$ represents for the horizontal low- vertical low subband in the lowest resolution. $HL_{cur}^{(i)}$, $LH_{cur}^{(i)}$, $HH_{cur}^{(i)}$ and $LL_{cur}^{(3)}$ have similar definitions except that they are from the current WB. (dx, dy) is the motion vector measured at the original spatial resolution. M and N are the dimensions of the WB with “%” being the modulus operation. In this work, they are both 16. The buffering of the reference frame is at least 10 frames when only previous picture is used as the reference for ME/MC. The WB in the reference frame/field with the smallest MAD is chosen as the matching WB for MC. The search window for MV of each WB is $[-16, +16]$ in the original pixel domain. For the motion vector (MV) of each WB, its west, north and northeast neighbors are used as predictors. The median of the motion information from these three predictors works as the prediction for the MV of the current WB. Then the difference between the motion information of the current WB and its prediction is calculated and coded using 8 contexts based on MQ coder. Here, the binarization and context design for MV coding from H.264 [5] is borrowed. Unary/ k th order Exp-Golomb (UEGk) binarization is constructed for motion vector differences (MVD). The horizontal and vertical component of each individual MVD is binarized according to the same rule, respectively. The context design of binarized string is same with that in H.264 [5]. Besides the UEGk binarization/ MQ coding approach, Witten’s algorithm [6] can also be used to code the motion information [2].

III. EXPERIMENTAL RESULTS

Experimental materials are in NTSC format with dimension 704×480 . This dimension has been chosen initially for the convenience of 3 to 5 level DWT. In this work, 3 level DWT is adopted. For the experiment, pictures are coded in IPIPIPE structure for all coders.

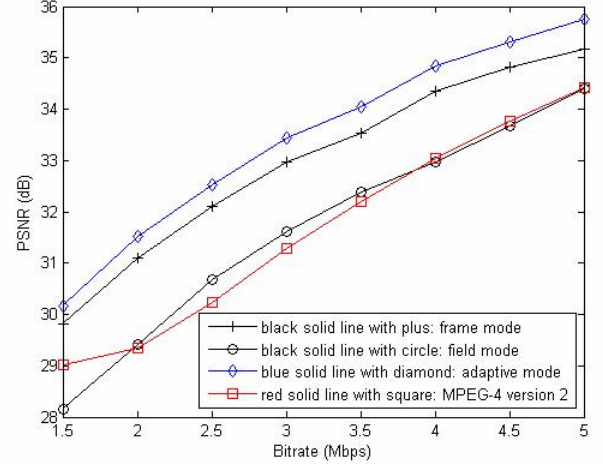


Fig. 1 Average PSNR results for 8 seconds of Baloon&Pops: pure frame mode, pure field mode, adaptive switching mode coding of the proposed coder are compared with MPEG-4 v.2.

The PSNR results of Baloon&Pops [7] are illustrated in Fig. 1. While the PSNR values of the proposed coder in either frame or field mode outperform those of MPEG-4 v.2. The adaptive switching mode of the proposed coder has superior performance to frame or field mode and by at least 1dB higher than the MPEG-4 v.2 coder. The authors are aware that comparison with H.264 is important and this will be part of the future work.

REFERENCE

- [1] A.S.Lewis and G.Knowles, "Video Compression Using 3D Wavelet Transforms," *IEEE Electronics Letters*, vol. 26, pp. 396-398, 1990.
- [2] L.M.Mei, H.R.Wu, and D.M.Tan, "A Wavelet Codec for Interlaced Video," presented at Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics VPQM-07, Scottsdale, Arizona, U.S.A., 2007.
- [3] D.S.Taubman and M.W.Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2002.
- [4] H.W.Park and H.S.Kim, "Motion Estimation Using Low-Band-Shift Method for Wavelet-based Moving Picture Coding," *IEEE Transactions on Image Processing*, vol. 9, pp. 577-587, 2000.
- [5] D.Marpe, H.Schwarz, and T.Wiegand, "Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 620-636, 2003.
- [6] I.H.Witten, R.M.Neal, and J.G.Cleary, "Arithmetic Coding for Data Compression," *Communications of The ACM*, vol. 30, pp. 520-540, 1987.
- [7] VQEG sequence "src13_ref_525". The index of VQEG sequences is available on <http://media.xiph.org/vqeg/TestSequences/ThumbNail/>.