DIRECTIONAL SPATIAL I-BLOCKS FOR THE MC-EZBC VIDEO CODER

Yongjun Wu and John W. Woods ‡

Center for Next Generation Video
Rensselaer Polytechnic Institute
Troy, NY 12180-3590, USA

ABSTRACT

In subband/wavelet interframe video compression, motion-compensated coders make use of estimated motion paths to compress the data. When these motion paths are not valid, due to either occlusion or difficulty in their estimation, artifacts can sometimes be created in the embedded lower frame rate video. Since these coders are intended for scalable application, it is important that the lower frame rate video be as free of artifacts as possible. The incorporation of I-blocks in the context of hierarchical variable size block matching (HVSBM) allows for inevitable occasional poorly connected motion blocks. Directional spatial interpolation/prediction can then be employed to minimize such a block's energy in a temporal high frame. Results are provided for the MC-EZBC coder that has been recently under investigation at MPEG.

1. INTRODUCTION

In the state of art scalable video coder MC-EZBC [1], hierarchical variable size block matching (HVSBM) is first employed to estimate the motion between adjacent frames. The motion estimation is bi-directional, using one previous reference frame and one future reference frame. For a given block, if a good match is not found in the future frame, we search in the past frame, and the better of the two matches is chosen as the reference block. If this bi-directional match results in too many unconnected pixels, further temporal analysis is stopped for this frame pair based on a specified threshold value. After successful bi-directional HVSBM, the full motion vector quad-tree is pruned back in the sense of rate-distortion optimization, with prediction error (MAD) as the distortion criteria.

Temporal filtering is then applied along the motion trajectories, allowing for efficient temporal decorrelation of sequences with local trackable motion. Since there are inevitable areas with occlusion or irregular motion, we classify the pruned blocks into three categories in MC-EZBC as shown in Fig. 1: DEFAULT blocks with motion vector between current frame A^t and next frame B^t, for which we do a lifting predict step in A^t and an update step for "connected" blocks in B^t. REVERSE blocks with motion vector between A^t and B^{t-1}, where we do motion compensation from corresponding blocks in B^{t-1}. P-BLOCKs with motion vector between A^t and B^t, for which we do a predict step in A^t but omit the update step for "unconnected" pixels in B^t. These P-BLOCKs come from multiply connected pixels as well as singly connected pixels with an excessively large a prediction error, as determined by a specified threshold (currently $\alpha = 0.5$) between the original block's variance and the motion-compensated block's MSE.

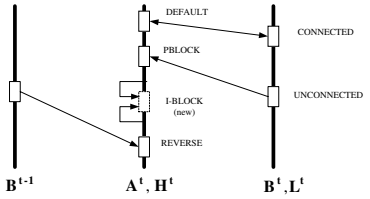

Fig. 1. Illustration of block modes in a high temporal frame for MC-EZBC. H^t is temporally aligned with A^t, and L^t is temporally aligned with B^t. The I-BLOCK only employs spatial interpolation/prediction

In the set of P-BLOCKs and REVERSE blocks, quite a few are poorly matched. These poorly matched blocks have high energies and produce block boundaries in the high temporal frames that reduce the compression efficiency. However, instead of interframe motion compensation, we can try to predict those blocks with size 8x8 or 4x4 from their spatial neighbors in the same frame. This does not only reduce their energy, but saves significantly on the motion vector rate, especially important at low video bit rates. We introduce the concept of *directional I-blocks* and call the new spatially interpolated/predicted block an I-BLOCK.

In this paper, we describe a method of directional spatial interpolation/prediction for directional I-blocks [2]. The included experimental results for a set of CIF sequences shows that directional I-blocks can provide up to 1.76 dB

‡Corresponding author. voice: (518) 276-6019; fax: (518) 276-6261; e-mail: woods@ecse.rpi.edu

mean PSNR gain (defined in section 3) at low bit rates for MC-EZBC.

2. DIRECTIONAL SPATIAL INTERPOLATION/ PREDICTION

2.1. I-BLOCK CANDIDATES

After pruning in bi-directional HVSBM, there are three kinds of blocks with variable size in a high temporal frame: DEFAULT, P-BLOCK, and REVERSE. Since P-BLOCKs are classified partially from DEFAULT blocks with a threshold $\alpha = 0.5$ between the original block's variance and motion-compensated block's MSE, they can include poorly matched blocks between A^t and B^t. We regard them as potential candidates for I-BLOCKs. Similarly we also look for I-BLOCK candidates from the set of REVERSE blocks.

2.2. DETECTING I-BLOCKS

For the two kinds of candidates in sizes 8x8 or 4x4, we try to interpolate/predict them from their spatial neighbors with the nine modes of H.264/MPEG [2]. Fig. 2 shows the mode down-left interpolation/prediction for a 4x4 block from its neighbors A, B, C, E, F and G. Neighbors A, H, and G are available for all modes if they are in the frame due to the quad-tree scan order employed in our HVSBM. The other neighbors are available, if their quad-tree scan order is before this block, or if they are not candidates for I-BLOCK, or if their block sizes are not 8x8 and 4x4. In spatial interpolation/prediction, we employ a "nearest neighbor" rule, i.e. as in Fig. 2: if neighbor C is available, we do not use neighbor B. According to the availability of the neighbors, we switch from spatial interpolation to spatial prediction adaptively. For example, when both neighbors A and G are available, linear interpolation from neighbors A and G is employed for the upper-left part of a 4x4 block. When only one of these neighbors is available, prediction from that neighbor is employed for the upper-left part.

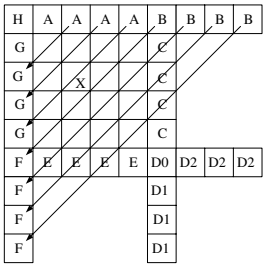

Fig. 2. Down-left interpolation/prediction for 4x4 block from neighbors A, B, C, E, F, and G

Generally, if there is some texture in poorly matched blocks, and the texture is approximately in one direction, we can find a suitable spatial interpolation/prediction mode for it. Actually, each mode can be regarded as texture interpolation/prediction with the direction interval 22.5^o except for the DC mode.

In order to fully use the neighbor information for a candidate block, we need to employ a *two-sweep* procedure for detecting I-BLOCKs. In the first sweep, for candidate block X, its block neighbors may also be I-BLOCK candidates. Those neighbors are not available for spatial interpolation/prediction in the first sweep. After the first sweep, whether candidates need spatial interpolation/prediction is determined according to whether the minimum MSE from spatial interpolation/prediction is less than that from motion compensation. Hence there may be more neighbors available for candidate X at that time. We check the nine modes again for the candidates employing spatial interpolation/prediction. As expected and from our experiment, almost all the MSEs from spatial interpolation/prediction decrease when more neighbors are available.

This two-sweep procedure is also critical for the needed consistency between the decoder and the encoder for spatial interpolation/prediction.

2.3. BLOCK PROCESSING ORDER IN MCTF

After the above detection procedure for I-BLOCKs, there are four kinds of blocks in a high temporal frame. We process the four kinds of variable size blocks in the motion-compensated temporal filter (MCTF) of MC-EZBC in the order shown in Fig. 3. At the decoder, this order is just reversed. This processing order ensures that I-BLOCKs use original data in their neighbors for spatial interpolation/prediction in the high temporal frames at the encoder, while at the decoder reconstructed data from the neighbors is used.

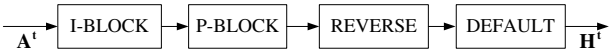

Fig. 3. Block processing order in the MCTF of MC-EZBC

For an I-BLOCK, the residual pixel value in a high temporal frame will be,

$$H^t[m,n] = \frac{1}{\sqrt{2}}(A^t[m,n] - \bar{A}^t[m,n]) \qquad (1)$$

where $\bar{A}^t[m,n]$ is the spatial interpolation/prediction value from the block's neighbors using some spatial mode with minimum MSE. For the decoder, the procedure is just reversed.

2.4. QUANTIZATION NOISE ANALYSIS

Using the same quantization noise model and the reconstruction procedure as in [3], for the pixels in DEFAULT, P-BLOCK, and REVERSE blocks, the resulting quantization noise is given by equations (2), (3) and (4) respectively:

$$\sigma^2_{A^t_C} = \sigma^2_{B^t_C} = \frac{1}{2}(\sigma^2_{L^t_C} + \sigma^2_{H^t_C}), \quad (2)$$

$$\sigma^2_{A^t_I} = 2\sigma^2_{H^t_I} + \sigma^2_{B^t}, \quad (3)$$

$$\sigma^2_{A^t_R} = 2\sigma^2_{H^t_R} + \sigma^2_{B^{t-1}}. \quad (4)$$

The quantization noise for "connected" pixels in frame B^t is also given by equation (2). The quantization noise for "unconnected" pixels in frame B^t is,

$$\sigma^2_{B^t_U} = \frac{1}{2}\sigma^2_{L^t_U} \quad (5)$$

Now we have an I-BLOCK for spatial interpolation/prediction. It's reconstruction procedure is

$$A^t[m,n] = \sqrt{2}H^t[m,n] + \bar{A}^t[m,n]. \quad (6)$$

Assuming the quantization noise is uncorrelated in different blocks, i.e. the quantization noise for $H^t[m,n]$ in (6), and that of its neighbor blocks used to produce $\bar{A}^t[m,n]$, it follows that the quantization noise variance for pixels in I-BLOCS is

$$\sigma^2_{A^t_T} = 2\sigma^2_{H^t_T} + \sigma^2_{\bar{A}^t}, \quad (7)$$

In most cases, I-BLOCKS are surrounded by P-BLOCKs and REVERSE blocks, or even other I-BLOCKs. So $\sigma^2_{\bar{A}^t}$ will probably be equal to $\sigma^2_{A^t_I}$, $\sigma^2_{A^t_R}$ or even the previous $\sigma^2_{A^t_T}$. Then, for equal quantization step sizes, the pixels in an I-BLOCK will probably have the largest quantization noise among the four kinds of blocks (at least the same as $\sigma^2_{A^t_I}$ or $\sigma^2_{A^t_R}$ when surrounded by DEFAULT blocks).

This means that in a statistical sense, the more spatial interpolated/predicted pixels a frame contains, the worse its PSNR can be. This may be the reason why there can be a slight 0.01-0.03 dB mean PSNR loss at bit rates above 1024 kbps where the reduction in motion vector bits and temporal high frame energy is not dominant compared to the negative effect of quantization noise. If optimized quantization [3] is employed, this slight mean PSNR loss at high rates should be lessened. This is a future topic.

3. EXPERIMENTAL RESULTS

We test directional I-block with a set of CIF sequences: *canoa*, *football*, *mobile*, *stefan* and *foreman* based on the version of MC-EZBC with YUV-HVSBM [4] and on that version with Y only HVSBM. Here YUV-HVSBM means the motion is estimated from all color components. For both versions, we found with directional I-block there is little gain at high bit rates, but for low rates below 512 kbps, the PSNR gain is significant and increases with decrease in bit rate, up to 1.76 dB for *football* at 256 kbps. This result comes from the fact that with directional I-block we can reduce the bits spent on motion vectors although we use a little more bits for the block map and interpolation/prediction modes. The bit reduction for motion vectors has a different effect from that for frame data, and is most important at low bit rates [5]. Table 1 shows PSNR comparisons for *football* in CIF format at various bit rates and the full frame rate (30 fps). Mean PSNR is defined as

(a)

(b)

Fig. 4. visual comparison of frame 3 (corresponding to original frame 6) at half frame rate and bit rate 256 kbps for canoa sequence in CIF format (a) with directional I-BLOCK (b) without directional I-BLOCK

Rate	Coder	Y	U	V	Mean
768	(1)	31.18	36.89	39.49	33.52
	(2)	31.09	36.79	39.46	33.44
512	(1)	29.31	35.20	38.01	31.74
	(2)	29.17	35.18	37.92	31.63
384	(1)	27.82	33.37	36.57	30.20
	(2)	27.59	33.25	36.33	29.99
256	(1)	24.41	29.86	33.95	26.91
	(2)	22.37	28.39	33.01	25.15

Table 1. PSNR (dB) comparison for *football* sequence in CIF format at various bit rates (kbps) and full frame rate (30fps) based on MC-EZBC with YUV HVSBM. Coder (1) with directional I-BLOCK, coder (2) without directional I-BLOCK.

Rate	Coder	Y	U	V	Mean
512	(1)	32.22	36.89	39.55	34.22
	(2)	32.11	36.93	39.55	34.15
384	(1)	30.79	35.83	38.62	32.93
	(2)	30.61	35.88	38.41	32.79
256	(1)	28.65	33.53	36.61	30.79
	(2)	28.45	33.39	36.53	30.62
192	(1)	27.12	32.13	35.42	29.34
	(2)	26.44	31.63	35.21	28.77

Table 2. PSNR (dB) comparison for *football* sequence in CIF format at various bit rates (kbps) and half frame rate (15fps) based on MC-EZBC with YUV HVSBM. Coder (1) with directional I-BLOCK, coder (2) without directional I-BLOCK.

$$\text{Mean} = (4 \times \text{PSNRY} + \text{PSNRU} + \text{PSNRV})/6.$$

We also checked the effect of directional I-block on the embedded low frame-rate video. Table 2 shows a PSNR comparison for *football* at various bit rates and 15 fps. We can see PSNR improvement here too. Since at lower frame rates the frames are further apart in time, we can expect more unconnected pixels, i.e. potential candidates for an I-BLOCK.

Fig. 4 provides a visual comparison for the third frame (corresponding to original frame 6) at half frame rate and bit rate 256 kbps for *canoa*. We can see that directional I-block can help to reduce some block artifacts in areas employing a directional I-block as expected. Comparison of the video is more dramatic.

The additional computation for the directional I-block is slight compared to that of the motion estimation, since the total number of pixels employing spatial interpolation/prediction averages less than 10% and we only need to check nine modes.

4. CONCLUSIONS AND FUTURE WORK

Use of directional I-blocks not only helps reduce the energy of poorly matched blocks in the high temporal frames, but also reduces the motion bit rate, which is needed for scaling down to low bit-rates.

Due to a higher quantization noise level for directional I-blocks, there may be a slight 0.01-0.03 dB average PSNR loss at high bit rates. We plan to extend optimized quantization [3] to include directional I-blocks to reduce this negative effect, in a rate-distortion optimization framework. Moreover, directional I-blocks can also help eliminate block boundaries around the blocks themselves. This is a promising effect that can be combined with overlapped block motion compensation(OBMC) [5] to result in further visual improvement.

5. REFERENCES

[1] P. Chen, K. Hanke, T. Rusert, and J. W. Woods, "Improvements to the MC-EZBC scalable video coder," Proc. IEEE Int. Conf. Image Process., vol. II, pp. 81-84, Sept. 2003, Barcelona.

[2] H.264/MPEG-4 Part 10: Intra Prediction at www.vcodex.com, I. Richardson.

[3] T. Rusert, K. Hanke, and J. Ohm, "Transition filtering and optimized quantization in interframe wavelet video coding," Proc. VCIP, vol. 5150, pp.682-693, July 2003, Lugano.

[4] A. Golwelkar, I. Bajic, and J. W. Woods, "Response to call for evidence on scalable video coding," ISO/IEC JTC1/SC29/WG11/M9723, July 2003.

[5] Y. Wu, R. A. Cohen, and J. W. Woods, "An overlapped block motion estimation for MC-EZBC," ISO/IEC JTC1/SC29/WG11/M10158, Oct. 2003.