MOTION HINTS BASED INTER-FRAME PREDICTION FOR HYBRID VIDEO CODING

Ashek Ahmmed #, Md. Jahangir Alam #, Aous Thabit Naman *, Mark Pickering #, and David Taubman *

School of Engineering and Information Technology, The University of New South Wales, Canberra, Australia.

* *School of Electrical Engineering and Telecommunications, The University of New South Wales, Sydney, Australia.*

***Abstract*—Experimental results and the latest standards have proved video coding systems with the ability to adapt the size and shape of the motion estimation area to the objects in the scene can outperform the traditional block-based video coding systems. In this paper, a segmentation-based coding strategy that employs bi-directional motion hints for inter-frame prediction is proposed. The appealing thing about motion hints is that they are continuous and invertible, even though the observed motion field for a frame will be discontinuous and non-invertible. The proposed scheme outperforms the rate-distortion performance of H.264/AVC reference by 1.1 dB and a bit rebate of 26.6% is achieved.**

***Index Terms*—motion hints, video coding, segmentation, motion estimation, H.264/AVC.**

I. Introduction

Conventional video coding depends on motion estimation and compensation to play crucial roles in granting high compression gains. The prediction of square or rectangular shaped motion blocks are provided by equally-shaped blocks that are to be found in the previously-decoded frames. With this approach the encoder and the decoder complexity remain minimal but compression inefficiency is incurred due to the unnatural division of the image to be coded. The use of variable sized blocks is an option to improve the final coding performance by trying to match the blocks to the objects in the scene. For example, the H.264/AVC standard [1] supports several types of block partitions from 4×4 to 16×16 pixels. More recently, the HEVC [2] standard allows prediction blocks sized up to 64×64 pixels. Careful partitioning of motion blocks in the vicinity of object boundaries represents a crude yet important way of segmenting the motion vector field into disjoint regions, with a smooth (typically constant) motion model within each block [3-6].

Building on the idea proposed in [7], Tagliasacchi *et al.* [8] proposed a motion estimation algorithm using a quadtree structure which produces a region based motion representation. A prune-merge scheme is used to segment the input image into regions. Their approach showed a gain of up to 0.6 dB and a rate rebate of 40%-50% at low bit-rates over the case that performs pruning only. A more flexible partitioning can be obtained via segmentation. In [9] an implicit block segmentation approach is proposed where segmentation is performed on the difference of the two predictors. This segmentation is based on the fact that for a 16×16 block each predictor may reduce the matching error non-uniformly inside that block. Their approach showed encouraging results for the *Foreman* sequence where illumination mismatches are not shown. Milani *et al.* [10] proposed a segmentation-based video coding system that partitions each frame into arbitrarily-shaped segments for a more effective motion compensation. Their scheme has been shown to outperform the rate-distortion performance of H.264/AVC of 2 dB with a reasonable increment of complexity in the encoder due to segmentation.

A novel approach was proposed in [11] that uses motion hints for inter-frame prediction. Motion hints provide a global description of motion over specific domains. Fundamentally this is related to the segmentation of foreground from background regions where the foreground and background motions are the motion hints. It has been shown that, with reasonably accurate motion, inter-frame predictions with good subjective quality and high PSNR can be generated [12].

Leveraging on the promising results shown by segmentation-based video coding and inter-frame prediction using motion hints, a segmentation-based coding strategy that finds bi-directional motion hints to perform motion compensated inter-frame prediction is proposed in this paper. The adopted coding strategy relies on the fact that motion hints permit identifying homogeneous regions of pixels in a frame that undergo similar motion. Therefore, it is possible to employ arbitrarily-shaped foreground and background regions rather than fixed or variable sized blocks for motion estimation and compensation and thereby reducing the amount of coded motion vectors enormously as well as approximating the real motion of objects in the scene more accurately. Experimental results show that the bit rate significantly reduces and the prediction PSNR improves.

The rest of this paper is organized as follows: in section II we describe the architecture of the proposed coder. Experimental results are reported in the following section. Finally, in section IV, we present our conclusions from these results.

II. Structure of the coding/decoding architecture

The adopted coding architecture has two main parts: the first part is a bi-directional motion hints based inter-frame prediction paradigm that performs the forward and backward foreground-background motion segmentations on the coded reference frames and generates the prediction of the current frame by projecting these foregrounds and backgrounds on to the current frame. The next step is mainly a decision making stage where the coder compares the predicted frame with the standard block-based coded version of the current frame in terms of the prediction PSNR. Based on the outcome of the comparison the coder decides what information needs to be sent to the decoder for that particular frame and then codes this information. In the following subsections, these parts are discussed briefly.

A. Bi-directional motion hints based inter-frame prediction

The first frame of each GOP is coded using the standard Intra mode of H.264/AVC [1] and transmitted to the decoder. The following frame, usually more than one time instance ahead, is coded using the temporally-predictive mode and the prediction residual signal and motion vectors are coded into a binary bit stream and transmitted to the decoder. Now these two already coded frames are used as references to predict the intermediate one(s) using a bi-directional

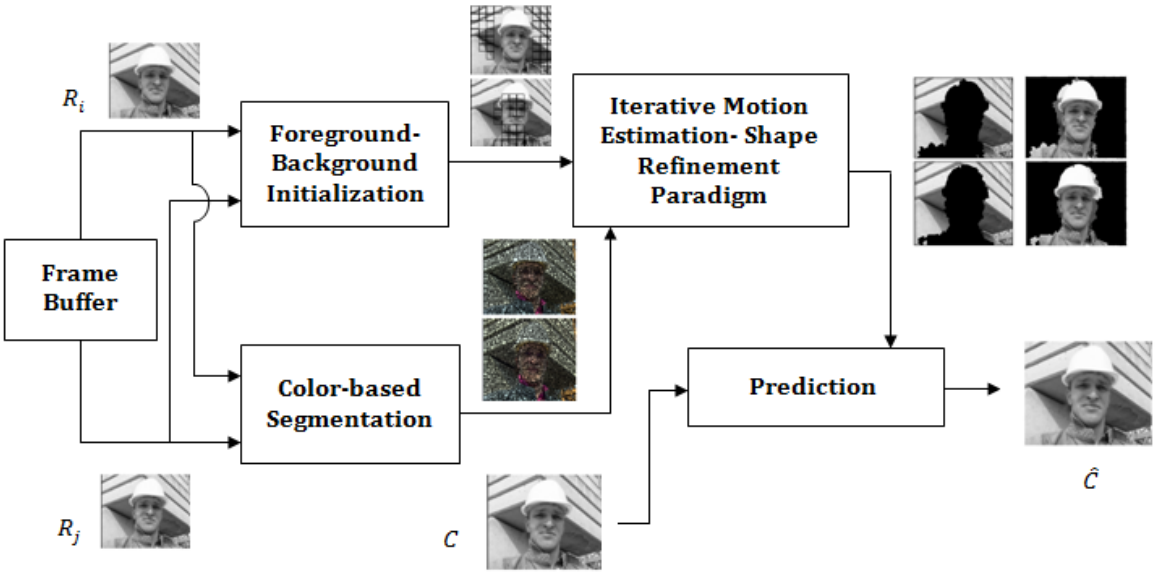


Fig. 1. Bi-directional motion hints compensated inter-frame prediction paradigm [13].

segmentation-based motion compensated prediction paradigm that employs motion hints where these hints are found by starting from an initial foreground-background segmentation and then refining them through successive motion estimation and compensation. In our earlier work [13] we discussed how this prediction paradigm works from the perspective of selecting a proper segmentation initialization strategy. Figure 1 shows a block diagram of the inter-frame prediction architecture. However, how the developed inter-frame prediction algorithm fits inside a coder needs to be investigated.

The two previously-coded frames which are used as references to predict the current frame are denoted by R_i and R_j respectively herein. A typical example of such frames are shown in Figure 1. The prediction paradigm starts by finding an estimate of the foreground-background shapes of R_j. In [13] the performance of four estimation strategies in terms of prediction PSNR was compared and the kurtosis-based initialization technique, which performed second best when bigger block sizes are used, is selected herein for generating the initial foreground-background segmentation with the aim of reducing the complexity of the encoder at the expense of providing slightly poorer prediction performance.

Having the initial foreground-background shapes, it is possible to estimate the initial values of the forward motion hints i.e. of $M_1^{(R_i \to R_j)}$ and $M_2^{(R_i \to R_j)}$. These motion hints take the form of a set of affine parameters that are used to warp the entire frame. The accuracy of these estimated hints improves with the improvement in the foreground-background shapes and vice versa. The precise location and boundary of the foreground-background regions can be estimated with the help of a color-based segmentation of R_j, as shown in Figure 1, that partitions R_j into super-pixels. The performance of each motion hint in compensating the motion within each super-pixel is investigated and a decision is made to include the super-pixel in the foreground or background segmentation mask. The predictions of R_j are generated by warping R_i with the forward motion hints one at a time. Along with the prediction errors, an estimate of the smoothness of the motion hint fields is used to regulate this optimization stage. The approach iterates between motion hints estimation and foreground-background shape refinement until the motion segmentation becomes stable. Once the foreground-background segmentation of R_j is achieved, the inverse of the forward motion hints and a color-based segmentation of R_i, shown in Figure 1, are used to initialize backward motion segmentation and thereby generate the foreground-background segmentation of R_i along with the backward motion hints $M_1^{(R_j \to R_i)}$ and $M_2^{(R_j \to R_i)}$. The same iterative 2-step strategy used to find forward motion segmentation is also used in this case. The forward and backward foreground-background segmentations are shown in Figure 1 as the output of the *Iterative Motion Estimation-Shape Refinement Paradigm*.

Next the prediction algorithm improves the estimated segmentations of R_i and R_j using the segmentation information available in both reference frames. At this stage the algorithm requires the knowledge of which region is the foreground of R_i and R_j. In the remainder of this example Region 2 is considered to be the foreground and Region 1 to be the background. The two foregrounds are corrected in such a way that they become related through the forward/backward foreground motion hints. It means for example, the corrected foreground mask of R_j is given by: $f_j^* = M_2^{(R_i \to R_j)}\left(f_i^*\right)$, apart from numerical approximations introduced by the warping process and any differences which might exist between $M_2^{(R_i \to R_j)}$ and $\left(M_2^{(R_i \to R_j)}\right)^{-1}$. Here f_i^* is the rectified foreground mask of R_i. The remaining parts of R_i and R_j are declared to be background masks.

Having the forward and backward foreground-background segmentations and four sets of motion hint fields, the prediction paradigm simply scales these motion fields to come up with versions of these fields from R_i to the current frame C and R_j to C. More specifically, the four sets of motion hint fields namely $M_1^{(R_i \to C)}$, $M_2^{(R_i \to C)}$, $M_1^{(R_j \to C)}$, and $M_2^{(R_j \to C)}$ are obtained by projecting the segmented foregrounds and backgrounds of R_i and R_j onto C respectively. These motion hints are then used to warp R_i and R_j to get four predictions, two apiece for the foreground and background of C.

Finally, by fusing these predictions through a weighting scheme $\widehat{C}$, the prediction of C is generated. Figure 1 shows an example of the current frame and its prediction produced by the prediction algorithm. For the decoder to generate the exact $\widehat{C}$ as of the encoder, it is sufficient to transmit these motion hints only. Receiving them the decoder can warp the decoded R_i and R_j, which are the exact copies R_i and R_j that the encoder has, by those hints and generate the desired $\widehat{C}$ and prevent any sort of drifting from happening. Although this approach reduces the bit rate when compared to the standard block-based coding approach, the savings is not that significant due to the fact that the motion hints are of large fractional precision. Downgrading this precision improves the bit rate but negatively impacts the prediction gain since warping is performed using a less accurate estimate of the true underlying motion from the references to the current frame.

To provide a trade off between the loss in prediction gain and the savings in bit rate, the motion hints are transformed from their affine model form to a translational model form described in [15]. As a way of explaining this new form for the motion hints, let's consider the situation of warping R_i by the 6-parameter motion hints field $M_1^{(R_i \to C)}$. The warping consists of transforming the image coordinates of R_i by applying the affine transformation specified in the motion hints field $M_1^{(R_i \to C)}$ as is done in the usual scenario. After this rather than getting the pixel intensities at the obtained pixel positions for the frame $M_1^{(R_i \to C)}\left(R_i\right)$ by interpolating the grey-levels of R_i at those pixel positions, at first the obtained pixel positions are quantized to restrict their fractional parts to be one of the four possible values $\{0, 0.25, 0.5, 0.75\}$. This ensures that the 3 translational motion vectors from R_i to $M_1^{(R_i \to C)}\left(R_i\right)$, specifically the top left, top right, and the centroid pixels' translational motion vectors [15], are of quarter pixel accuracy. And these motion vectors referred to as the corner motion vectors herein are then transmitted to the decoder rather than the associated 6-parameter motion hints field itself which is of high fractional precision. The decoder generates an estimate of the original motion hints field $M_1^{(R_i \to C)}$ from the received corner motion vectors and the already available information i.e. the top left, top right and centroid pixel positions of R_i. However, this deduced motion hints field $\widehat{M}_1^{(R_i \to C)}$ is an estimate of the actual motion hints field $M_1^{(R_i \to C)}$, which the encoder possesses. The reason behind this drifting between the encoder and the decoder is that the corner motion vectors are downgraded to have quarter pixel accuracy so using them the true motion hints field $M_1^{(R_i \to C)}$ can not be recovered at the decoder. To prevent any drifting from happening the encoder mimics the decoder i.e. it warps R_i by $\widehat{M}_1^{(R_i \to C)}$ and then uses a bi-cubic interpolation to find the intensities at the obtained pixel positions. The obtained prediction in this way is different from the prediction shown in Figure 1 but in experimental evaluation it has been found that it matches $\widehat{C}$ more closely than any prediction carried out with rounded motion hints i.e. with significantly less accurate motion hints. Before coding the corner motion vectors, they are multiplied by 4 to have the advantage of coding integers. Experimental results show a significant reduction in bit rate over the traditional block-based prediction approach.

B. Comparison with the H.264/AVC coded frame

Once the prediction of the current frame from the bi-directional motion hints compensated inter-frame prediction paradigm is available, it is then compared with the standard block-based coded version of the current frame in terms of the prediction PSNR. If the H.264/AVC coded B-frame has higher PSNR, the associated prediction residual and motion vectors are coded into a binary bit stream and transmitted to the decoder. However, if the prediction from the motion hints based approach has higher PSNR, in this case no prediction residual information is sent to the decoder rather only the motion hints are transmitted. The affine motion hints fields are coded by transmitting the corner motion vectors for each of the four motion hints fields required to generate the motion hints based prediction. In both possible cases a single bit is sent to the decoder to inform which of the two available $\widehat{C}$'s it should use. For example, if the signalling bit is 1 then the decoder understands that it should use the bi-directional motion hints compensated prediction frame rather than the standard block-based coded B-frame. Then the decoder generates the four predictions of the foregrounds and backgrounds of the current frame by using the previously decoded P-frames and the corner motion vectors information which are divided by 4 at first . And finally fuses them to get the bi-directional motion hints compensated B-frame of the encoder.

Next we investigate the performance of the proposed coder on the QCIF sequences *Foreman*, *Stair* [14] where a person is walking and another person is following him and recording him with a hand-held camera and *Hand held Mobile phone* where a person mimics video conferencing on a mobile phone by recording himself talking and at the same time moving while the mobile phone is in his right hand.

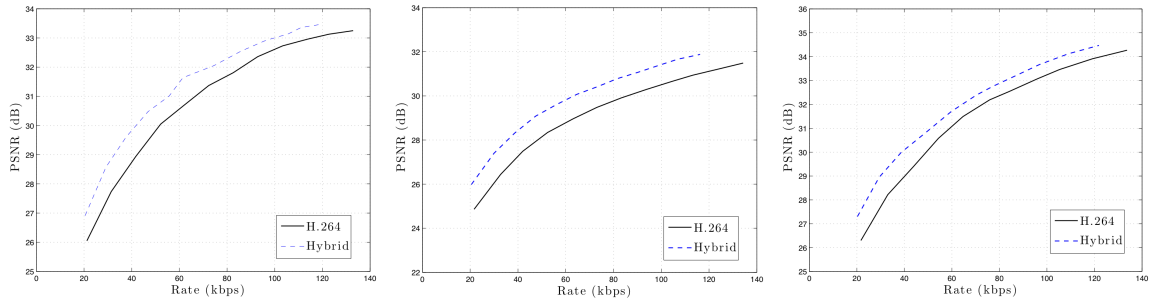
III. Experimental Results

To evaluate the performance of the hybrid video coder, each input sequence is partitioned into GOPs of 12 frames (IBBPBBPBBPBB) and coded by the JM software. We used 200 frames from each of the sequences to execute the tests. The coded (I,P) or (P,P) pair of frames are fed into the bi-directional motion hints compensated inter-frame prediction paradigm to generate predictions for the intermediate frames. In the hybrid coder these predictions are compared with their H.264/AVC coded counterparts and only the winners are kept in the frame buffer. For a motion hints based prediction only the corner motion vectors of each of the two foreground predictions and two background predictions are transmitted while for the standard block-based prediction, the residual and motion vectors are transmitted to the decoder.

The performance improvement between the coders is measured in terms of the Bjontegaard metric. Figure 2 shows the Rate-Distortion curves for the test sequences. For the *Foreman* sequence, the average gain in PSNR from the hybrid coder over the reference coder is 0.86 dB and a rate rebate of 19.7% is achieved. While for the *Stair* sequence the savings in bit rate jumps to 26.6% with an average coding gain of 1.1 dB. On the other hand, for the *Hand held Mobile Phone* sequence a rate rebate of 17.9% is achieved with a gain of 0.86 dB. Figure 3 shows the fraction of motion hints predicted B-frames used by the hybrid coder at different bit-rates for the three test sequences. The more this number is at a given bit rate the higher the bit rebate would be. As the bit rate increases this number tends to fall down but still remains above a reasonable threshold therefore although with increasing bit rate the gain in prediction PSNR decreases, some coding gain is still achievable.

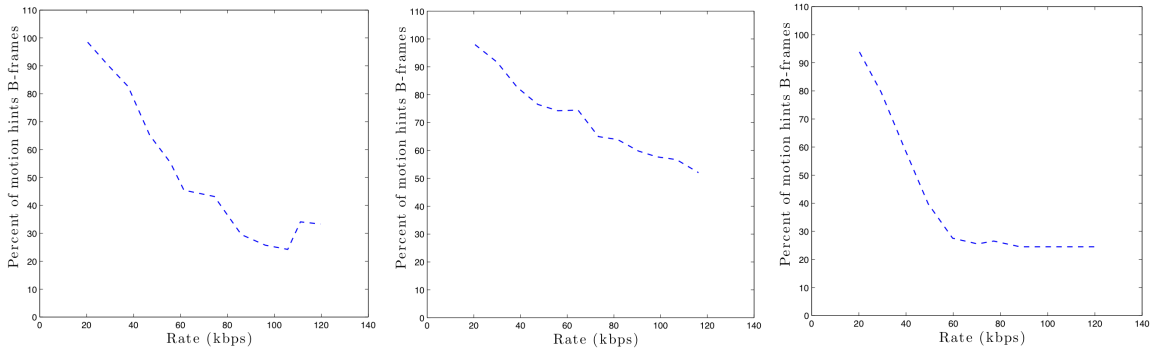
IV. Conclusions

In this paper, we have presented a segmentation-based hybrid video coder that incorporates a bi-directional motion hints based inter-frame prediction strategy. The coder compares the motion hints based B-frame with the standard block-based B-frame in terms of prediction PSNR and transmits only the corner translational motion vectors to the decoder if the motion hints based B-frame comes out as the winner of the comparison. Experimental results show an overall



(a) *Foreman* (b) *Stair* (c) *Hand held mobile phone*

Fig. 2. PSNR vs. bit rate of different coding strategies for different QCIF sequences.



(a) *Foreman* (b) *Stair* (c) *Hand held mobile phone*

Fig. 3. Percent of motion hints compensated B-frames used by the hybrid coder at different bit rates.

significant improvement both in prediction PSNR (up to 1.1 dB) and bit rebate (up to 26.6%).

REFERENCES

[1] Joint Video Team (JVT) of ISO/IEC MPEG and IUT-T VCEG, "Joint final committee draft (JFCD) of joint video specification (ITU-T Rec. H.264-ISO/IEC 14496-10 AVC)," in *Joint Video Team, 4th Meeting*, Klagenfurt, Germany, July 2002.

[2] T. Davies, K. R. Andersson, R. Sjberg, T. Wiegand, D. Marpe, K. Ugur, J. Ridge, M. karczewicz, P. Chen, G. Martin-Cocher, K. McCann, W. J. Han, G. Bjontegaard, and A. Fuldseth, "Suggestion for a test model," in *Joint Collaborative Team on Video Coding (JCT-VC), 1st Meeting*, Dresden, Germany, Apr. 15-23, 2010.

[3] A. A. Muhit, M. R. Pickering, and M. R. Frater, "A fast approach for geometry-adaptive block partitioning," *Picture Coding Symposium (PCS)*, 2009, pp. 413-416.

[4] A. A. Muhit, M. R. Pickering, and M. R. Frater, "Motion compensation using geometry and an elastic motion model," *IEEE Int. Conf. Image Processing (ICIP)*, Nov 2009, pp. 621-624.

[5] R. U. Ferreira, E. M. Hung, R. L. de Queiroz, and D. Mukherjee, "Efficiency improvements for a geometric-partition-based video coder," *IEEE Int. Conf. on Image Processing (ICIP)*, 2009.

[6] R. Mathew and D. S. Taubman, "Scalable modeling of motion and boundary geometry with quad-tree node merging," *IEEE Trans. CSVT*, vol.21, no.2, pp.178-192, Feb. 2011.

[7] R. D. Forni and D. Taubman, "On the benefits of leaf merging in quad-tree motion models," Proc. *IEEE Int. Conf. Image Processing*, vol. 2, pp. 858-861, September 2005.

[8] M. Tagliasacchi, M. Sarchi, and S. Tubaro, "Motion estimation by quadtree pruning and merging," *IEEE Int. Conf. Multimedia Expo (ICME)*, July 2006, pp. 1861-1864.

[9] J. Kim, A. Ortega, P. Yin, P. Pandit, and C. Gomila, "Motion compensation based on implicit block segmentation," *IEEE Int. Conf. on Image Processing (ICIP)*, 2008, pp. 2452-2455.

[10] S. Milani and G. Calvagno, "Segmentation-based motion compensation for enhanced video coding," *IEEE Int. Conf. on Image Processing (ICIP)*, 2011, pp. 1685-1688.

[11] A.T. Naman, D. Edwards, and D. Taubman, "Efficient communication of video using metadata," *18th Proc. IEEE Int. Conf. Image Proc.* 2011, pp. 589592, September 2011.

[12] A.T. Naman, R. Xu, and D. Taubman, "Inter-frame prediction using motion hints," *20th Proc. IEEE Int. Conf. Image Proc.*, September 2013.

[13] A. Ahmmed, R. Xu, A.T. Naman, M. J. Alam, M. Pickering, and D. Taubman, "Motion segmentation initialization strategies for bi-directional inter-frame prediction," *IEEE Int. Workshop on Multimedia Signal Process.*, September 2013.

[14] G. Zhang, J. Jia, W. Hua, and H. Bao, "Robust bilayer segmentation and motion/depth estimation with a handheld camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 603617, 2011.

[15] H. Lakshman, H. Schwarz, and T. Wiegand, "Adaptive motion model selection using a cubic spline based estimation framework," *17th Proc. IEEE Int. Conf. Image Proc.*, 2010, pp. 805-808.