MOTION COMPENSATION BASED ON IMPLICIT BLOCK SEGMENTATION

Jae Hoon Kim, Antonio Ortega

Signal and Image Processing Institute
Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089-2564

Peng Yin, Purvin Pandit and Cristina Gomila

Corporate Research Princeton, Thomson Inc.
2 Independence Way
Princeton, NJ, 08540

ABSTRACT

Block-based motion and disparity compensation are popular techniques to exploit correlation between video frames. Block sizes used for compensation can be chosen to achieve a good trade-off between signaling overhead and prediction accuracy. However, motion field boundaries correspond to objects having arbitrary shapes; this limits the accuracy of block-based compensation, even when small block sizes are chosen. In this paper we seek to enable compensation based on arbitrarily-shaped regions, while preserving an essentially block-based compensation architecture. To do so, we propose tools for implicit block-segmentation and predictor selection. Given two candidate block predictors, segmentation is applied to the difference of predictors. Then a weighted sum of predictors in each segment is selected for prediction. Simulation results show improvements in rate-distortion (RD) performance, as compared to the standard quad tree approach in H.264/AVC.

Index Terms— Video coding, motion search, hierarchical quad-tree, H.264/AVC, segmentation

1. INTRODUCTION

Exploiting inter-frame correlation via motion estimation and compensation is key in achieving high video compression efficiency. Block-based motion compensation provides a good balance between prediction accuracy and rate overhead. Clearly, blocks of pixels are not guaranteed to have uniform displacement across frames. For video sequences this is the case if an object boundary exists in a block and pixels which belong to different objects move in different ways. In stereo or multi-view sequences objects in different depths have different disparities and occlusion effects. This makes disparity search difficult and reduces coding efficiency in cross-view prediction.

Numerous approaches have been proposed to provide more accurate motion compensation by providing different prediction for different regions in a macroblock. Examples include techniques used in H.264/AVC video coding standards [1] or the hierarchical quad-tree (QT) approach [2]. In these methods a macroblock is split into smaller blocks and the best match for each block is searched. As the number of blocks in a macroblock increases, overhead increases and distortion between original and the match decreases. Therefore, there is a minimum rate-distortion point and the best block mode is decided by Lagrangian tool. To increase the matching capability by square or rectangular block shape in QT, geometry based approach (GEO) is proposed in [3, 4]. A block is split into two smaller blocks called wedges by a line described by slope and translation parameters. The best parameters and matching wedges are searched together. Although GEO captures object boundaries better than quad tree, it is still limited to be a straight line. In [5], an object based motion segmentation method is proposed to solve the occlusion problem. To capture different motions in a block, motion vectors from neighboring blocks are copied after block segmentation. To avoid transmitting segmentation information, previously encoded frames at $(t-1)$ and $(t-2)$ are used to estimate segmentation for the current frame at (t).

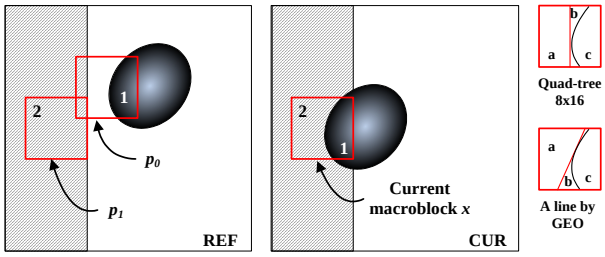


Fig. 1. Example of block matching by segmentation

In this work, we present a framework for implicit block segmentation to increase prediction quality. Implicit block segmentation is obtained based on the predictors from previously encoded frames as in [5]. However, segmentation is applied to the difference of two predictors, rather than directly to the predictor itself. Also, unlike in [5] motion vectors are explicitly transmitted to signal the location of chosen predictors and the encoder searches for the best combination of predictors. We use 16×16 macroblocks, which are assumed to be relatively small relative to typical objects in the scene, so that in many cases at most two objects move with different displacements at the boundaries [5]. Although distortion can be reduced as the number of predictors increases, the overhead required for motion/disparity vectors and for identifying the selected predictor for each segment also increases with the number of predictors. While the number of predictors can be optimally chosen based on R-D cost similarly to hierarchical quad-tree, in this work for simplicity we choose the maximum number of predictors to be two.

In Section 2, details of block segmentation and its implementation within an H.264/AVC architecture are described. In Section 3, simulation results for temporal and cross-view sequences are shown. In Section 4, conclusion follows.

2. BLOCK BASED SEGMENTATION

Fig. 1 shows an example of block motion compensation between current and reference frame. In the current block, we have two ob-

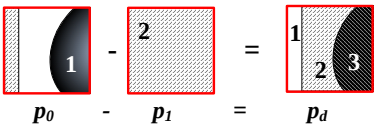


Fig. 2. Definition of predictor difference $\bar{p}_d$ in Fig. 1.

jects which are separated by a smooth boundary. Let us assume that the correct matches of each object can be found as a base predictor ($\bar{p}_0$) and a enhancement predictor ($\bar{p}_1$) as shown in the reference frame. In Fig. 2, we depict the difference between the two predictors, $\bar{p}_d = \bar{p}_0 - \bar{p}_1$. In region 1 of $\bar{p}_d$, the absolute difference of pixel values is small because $\bar{p}_0$ and $\bar{p}_1$ come from the same object and both $\bar{p}_0$ and $\bar{p}_1$ will estimate original with small error. Therefore, the difference in residual error when using the two predictors (i.e., $|\bar{x} - \bar{p}_0| - |\bar{x} - \bar{p}_1|$) will tend to be small. In regions 2 and 3 of $\bar{p}_d$, $\bar{p}_1$ and $\bar{p}_0$, respectively, provide the best match. Thus, the absolute difference between the two predictors will tend to be large, and we similarly would expect that the differences in residual error after prediction will be large.

For each region several scenarios are possible. In the area where $|\bar{p}_d|$ is small, because the two predictors are similar we have that either i) both predictors provide a good match or ii) the residual error is large with respect to both predictor and choosing one of the predictors over the other will not lead to significant improvements. Instead, in areas where $|\bar{p}_d|$ is large, either i) only one of the two predictors provides a good match, or ii) a combination of both predictors may lead to a better matching performance. Clearly, choosing the "right" predictor among the two available choices is more important for regions where $|\bar{p}_d|$ is large; it is in these regions where signaling a predictor choice can lead to a more significant gain in prediction performance.

For the original macroblock signal $\bar{x}$, QT and GEO find the best predictor for each segment respectively. Therefore, although the correct matches for each object are given as $\bar{p}_0$ and $\bar{p}_1$, neither QT nor GEO finds correct match without significant prediction error in region b in Fig. 1 because object boundary is not aligned by a straight line.

2.1. Implicit Block Segmentation (IBS)

Assume two predictors are available for a given macroblock (i.e., two 16×16 blocks from neighboring frames). These two predictors have been chosen by the encoder and their position will be signaled to the decoder. The optimal segmentation for the purpose of prediction would be such that each pixel in the original macroblock is assigned to whichever predictor, $\bar{p}_0$ or $\bar{p}_1$, provides the best approximation. However this cannot be done implicitly (without sending side information) since the decision depends on the original block itself. Based on our previous observations about the expected gain depending on the differences between predictors, we apply segmentation to the block of predictor differences, $\bar{p}_d$. Due to the noisy characteristics of predictor differences, edge based segmentation methods do not detect simple boundaries efficiently in 16×16 macroblocks. In this work, 1-D K-means clustering [6] is used as a basic segmentation algorithm. N_0 centroids are initialized at the uniform distance between maximum and minimum value of $\bar{p}_d$. Maximum run is set to 20. After K-means clustering, disconnected pixels exist which belong to the same segment because spatial connectivity is not considered in 1-D K-means clustering. A two step post-processing is

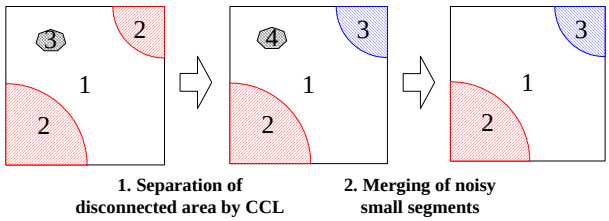


Fig. 3. Post-processing after segmentation with segment index. First, disconnected segment 2 is classified as different segment increasing the number of segment N from 3 to 4. Second, segment 4 is merged into segment 1 decreasing N to 3 again.

applied to take spatial information into account. First, using connected component labeling [7], disconnected pixels assigned to the same segment are classified into different segments. Second, to prevent noisy segments, if the number of pixels in a segment is smaller than N_{pix}, it is merged into the neighboring segment that has the minimum segment-mean difference with current segment. Fig. 3 depicts this post-processing. Note that the number of segments depends on the disparities between base and enhancement predictors. In this work, N_0 and N_{pix} is set to be 2 and 10, experimentally.

For each segment k in $\bar{p}_d$, the optimal predictor $\hat{x}_k$ can be calculated as a weighted sum of base and enhancement predictors when original $\bar{x}$ is known. If scalar weights α_k and β_k are applied to all pixels in segment k of $\bar{p}_0$ and $\bar{p}_1$, sum of squared difference (SSD) for the segment k is

$$SSD_k = ||\bar{x}_k - \hat{x}_k||^2 = ||\bar{x}_k - (\alpha_k \bar{p}_{0,k} + \beta_k \bar{p}_{1,k})||^2 \quad (1)$$

$\bar{p}_{0,k}$ and $\bar{p}_{1,k}$ specifies the pixels of $\bar{p}_0$ and $\bar{p}_1$ belonging to segment k. By setting to zero the gradient of eq. (1) with $\alpha_k + \beta_k = 1$, optimal weights can be found as

$$\begin{aligned} \alpha_k &= \frac{-(\bar{p}_{1,k} - \bar{x}_k) \cdot \bar{p}_d}{||\bar{p}_d||^2} \\ \beta_k &= \frac{(\bar{p}_{0,k} - \bar{x}_k) \cdot \bar{p}_d}{||\bar{p}_d||^2} \end{aligned} \quad (2)$$

Because the optimal α_k is calculated using information from the block to be encoded, the chosen value has to be signaled. For 16×16 blocks, this signaling overhead may not be justified given the overal reductions in residual error. In order to limit the overhead, in this work, weights are selected from a predefined set $W = \{(1,0), (0,1), (\frac{1}{2}, \frac{1}{2})\}$, corresponding to using $\{\bar{p}_0, \bar{p}_1, \frac{1}{2}(\bar{p}_0 + \bar{p}_1)\}$ for prediction, respectively. Thus a weight index with only three values $\{0, 1, 2\}$ has to be signaled[1]. In summary, prediction for the block to be encoded is achieved by signaling the two predictors, $\bar{p}_0$ and $\bar{p}_1$, and the weights to be used for each segment, w_k. The segmentation itself is generated by encoder and decoder in the same manner from the decoded predictors, so that there is no need for side information to be sent.

Since prediction is performed by combining two predictors using our proposed IBS technique, there is no guarantee that one could obtain the best matching pair of predictors by search for each predictor individually using standard residual energy metrics based on the

[1]Note that it is easy to extend this framework by including additional weights in W. With binary arithmetic coding or variable length coding of weight indices, a given weight will be chosen only if it leads to gains in an RD sense.

whole 16×16 block. In theory one would have to search for *pairs* of predictors, i.e., for each base predictor candidate, it would be necessary to search all candidate enhancement predictors and choose the best one by computing the prediction residue *after segmentation and combined base/enhancement prediction*. This is illustrated in Fig. 4. This general approach would have significant complexity. Instead, we start by obtaining the top M base predictors using standard blockwise metrics. Then we perform a joint search for enhancement predictors for only those M base predictors. The cost functions for each step are described next.

2.2. Implementation within an H.264/AVC architecture

Implicit block segmentation is implemented in H.264/AVC reference codec - *JSVM 8.4*. Current inter block modes are extended inserting $INTER16\times16\_IBS$ between $INTER16\times16$ and $INTER16\times8$. RD optimization tool in H.264/AVC is applied to choose the best mode for each macroblock.

If the number of candidates in full search range is N_{fs}, there are N_{fs}^2 candidates for $INTER16 \times 16\_IBS$ when full search is applied to pairs of base and enhancement predictors. This is significantly higher than the number of search locations in the hierarchical QT, $N_{QT}N_{fs}$, where N_{QT} is the number of QT block modes and $N_{QT} \ll N_{fs}$. Instead of testing all candidates in search range for base predictor, a limited set of candidates is collected from the best matches of $INTER16 \times 16$, $INTER16 \times 8$, $INTER8 \times 16$, $INTER8 \times 8$ and original block segments. If original macroblock is segmented into N_{org} regions after post-processing, N_{org} best matches for the segment are found during $INTER16 \times 16$ motion search. Because duplicate candidates are removed, $(N_{org} + 9)$ is the maximum number of base predictor candidates.

To select the best predictor pair $(\bar{p}_0, \bar{p}_1)$ for $INTER16 \times 16\_IBS$, three different error metrics are used. For each base predictor candidate, the best complementary enhancement predictor is searched within search range as in Fig. 4. The first error metric is sum of absolute difference (SAD) used to decide the weight index w_k for the segment k. SAD of segment k is calculated for all weights in W and the weight index with minimum SAD is chosen. Second, in the selection of the best enhancement predictor for given base predictor, a simplified R-D cost, J, is defined as;

$$J = [\sum_{k=1}^{N} min_{w_k}\{SAD_k\}] + \sqrt{\lambda}NB + \sqrt{\lambda}MVcost(\bar{p}_1)$$

N is the number of segments in $\bar{p}_d$, B is the number of bits for weight index per segment defined as $B = log_2|W|$ and $MVcost(\bar{p}_1)$ is the motion vector cost of enhancement predictor $\bar{p}_1$. $MVcost(\cdot)$ and λ follows the definition of H.264/AVC.

For M base predictor candidates, equal numbers of matching enhancement predictors are found. Finally, RD cost of M base and enhancement predictor pairs are calculated and compared with RD costs of other block modes in H.264/AVC (RD mode decision). Encoded information in $INTER16 \times 16\_IBS$ includes reference indices and motion vectors for base and enhancement predictors as well as the weight indices for each segment. Weight indices are encoded by variable length code in R-D mode decision and binary arithmetic code in bit stream coding.

In Fig. 5, an example of predictor difference between base and enhancement predictor is shown, with its corresponding segment information. Predictor difference shown in Fig. 5 (a) is scaled to show the difference clearly. Note that the segmentation shown in Fig. 5 (b)

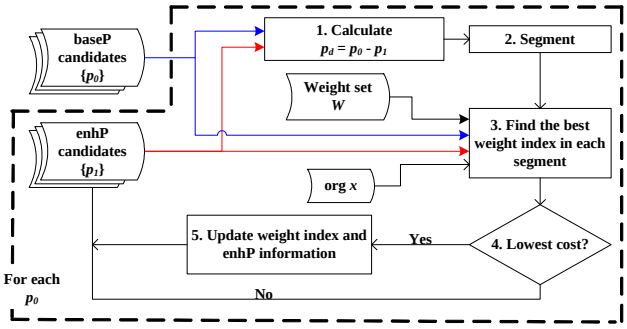


Fig. 4. Search loop of enhancement predictor for given base predictor

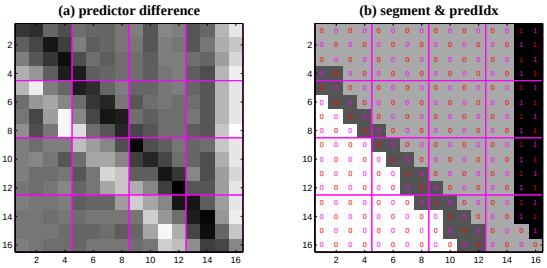


Fig. 5. (a) Predictor difference of base and enhancement predictor (b) segmentation from predictor difference and chosen weight index

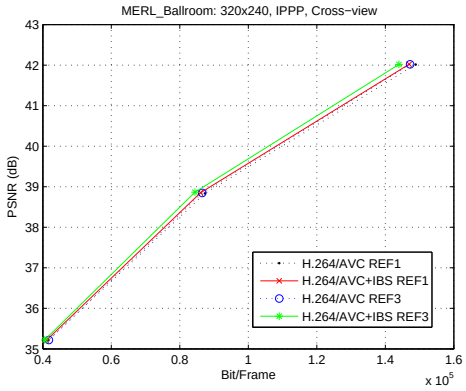
captures large predictor differences efficiently. Selected weight indices show that the top right part of the block uses enhancement predictor, while elsewhere the base predictor is used. Prediction using by IBS achieves 30% SSD reduction as compared with the best predictor by quad-tree in Fig. 5.

3. SIMULATION RESULTS

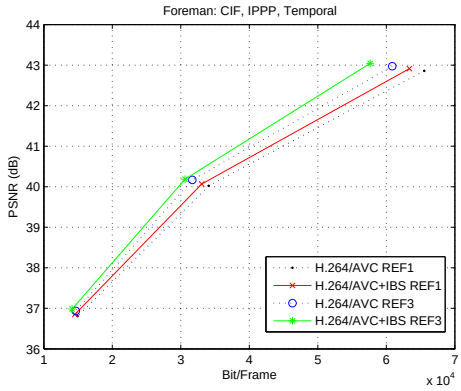
Both multi-view video (*MERL_Ballroom*, 320(w)x240(h)) and standard video sequences (*Foreman*, 352(w)x288(h)) are tested. In *MERL_Ballroom*, each anchor has 8 views coded IPPP PPPP and 2 anchors at different time stamps (0, 10) are tested. In *Foreman*, 15 frames are coded as IPPP. Encoding conditions of 'H.264/AVC' and 'H.264/AVC+IBS' are the same except that in 'H.264/AVC+IBS', $INTER16 \times 16\_IBS$ is tested as an additional inter block mode. $QP20, 24, 29$ are used with ±32 search range with quarter-pel and CABAC enabled. As can be seen in Fig. 6, 0.1-0.2 dB gains are achieved in *MERL_Ballroom* and 0.2-0.4 dB gains from *Foreman*. Note that gains by IBS increase with the number of references.

In Table 1, average distortions and bits are shown when IBS is the best mode during mode decision. Prediction quality improves by IBS as can be seen in the reduction of SSD_{pred}. This is translated into reduction in residual coding bits while SSD in reconstructed frame, SSD_{recon}, does not change significantly. Note that typically the bits needed signal motion vectors are reduced because only two predictors are used in IBS (while a QT approach could use more than two vectors). Extra bits are needed to signal weights when using IBS.

Gains are not encouraging in *MERL_Ballroom*. Firstly, due to the noisy background of *MERL_ Ballroom*, predictor difference results in noisy segments, which increases signaling bits for weight



(a) *MERL_Ballroom* with 1 and 3 reference



(b) *Foreman* with 1 and 3 references

Fig. 6. Comparison of 'H.264/AVC' and 'H.264/AVC+IBS'

Table 1. Comparison of data by QT and IBS from *MERL_Ballroom* and *Foreman*) with QP 20. Data is averaged for the macroblocks where IBS is the best mode from 14 P-frames in each sequence. $A \rightarrow B$ means 'best data by QT' $\rightarrow$ 'best data by IBS'. SSD_{pred} is the sum of squared difference between the original and predictor. Bit_{mv}, Bit_{res} and Bit_w are bits for motion/disparity vectors, residual and weight indices respectively.

Sequence	SSD_{pred}	Bit_{mv}	Bit_{res}	Bit_w
MERL_ Ballroom	13928→11538 *(17%)*	20→16 *(18%)*	433→409 *(5%)*	0→9.1
Foreman	3473→3078 *(11%)*	23→16 *(32%)*	159→145 *(9%)*	0→7.6

indices as shown in Table 1. Secondly, for implicit block segmentation, it is assumed that references are not corrupted or mismatches including illumination and focus do not exist between frames. As shown in [8], there *exist* illumination mismatches between frames in different views. When two different segments with non-zero DC level exist in a 4x4 or 8x8 DCT block, it increases high frequency components thus, residual coding bits increase. Also it may create artificial boundary within a block. Note that in Table 1, 17% reduction in SSD_{pred} is translated into only 5% reduction in residual bits in *MERL_Ballroom*. Combined with illumination compensation [8], the performance of IBS for cross-view prediction could be improved.

4. CONCLUSIONS

We proposed implicit block segmentation based on the predictors available at the decoder. From the observation that distortion can be reduced further where two predictors differ most, segmentation is applied to the predictor difference. Different weighted sums of predictors are selected for each segment and signaled to the decoder. Implementation in H.264/AVC shows encouraging results in *Foreman* where illumination mismatches are not shown. Combining IBS with mismatch compensation tools would increase the coding efficiency in cross-view prediction. Areas of future work include improvements to the segmentation strategy and efficient search techniques to allow searching for pairs of predictors.

5. REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Systems and Video Technologies*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[2] R. Shukla, P. Dragotti, M. Do, and M. Vetterli, "Rate-distortion optimized tree-structured compression algorithms for piecewise polynomial images," *IEEE Trans. Image Processing*, vol. 14, no. 3, pp. 343–359, Mar. 2005.

[3] O. Divorra Escoda, P. Yin, D. Congxia, and L. Xin, "Geometry-adaptive block partitioning for video coding," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2007, vol. 1, pp. I–657–I–660.

[4] E. M. Hung and R. L. De Queiroz, "On macroblock partition for motion compensation," in *Proc. of IEEE International Conference on Image Processing (ICIP)*, 2006.

[5] M. Orchard, "Predictive motion-field segmentation for image sequence coding," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 3, no. 1, pp. 54–70, Feb. 1993.

[6] D. Mount, "KMlocal: A testbed for k-means clustering algorithms based on local search Version: 1.7.1," *Dept of Computer Science at University of Maryland, http://www.cs.umd.edu/ mount/Projects/KMeans/*, 2005.

[7] F. Chang, C.J. Chen, and C.J. Lu, "A linear-time component-labeling algorithm using contour tracing technique," vol. 93, no. 2, pp. 206–220, February 2004.

[8] J. H. Kim, P. Lai, J. Lopez, A. Ortega, Y. Su, P. Yin, and C. Gomila, "New coding tools for illumination and focus mismatch compensation in multiview video coding," *IEEE Trans. Circuits Systems and Video Technologies*, vol. 17, no. 11, pp. 1519–1535, Nov. 2007.