A PRE-FILTERING APPROACH TO EXPLOIT DECOUPLED PREDICTION AND TRANSFORM BLOCK STRUCTURES IN VIDEO CODING

Yue Chen⋆,Kenneth Rose⋆, Jingning Han †, Debargha Mukherjee†*

⋆Department of Electrical and Computer Engineering, University of California Santa Barbara, CA 93106
†Google Inc., 1600 Amphitheatre Parkway, Mountain View, CA 94043
E-mail: ⋆{yuechen,rose}@ece.ucsb.edu, †{jingning,debargha}@google.com

ABSTRACT

Recent video coding techniques allow for decoupling of the transform block partition from that employed for prediction. For example, HEVC allows a transform block to overlap multiple prediction blocks. This paper is premised on the observation that in order to truly realize the potential of such enhanced flexibility, it is necessary to account for and mitigate considerable side effects due to stitching together independently predicted blocks, including the emergence of spurious high frequency components from sharp transitions across boundaries, which undermine the transform efficacy. The proposed solution involves an appropriately designed pre-filtering approach to mitigate boundary transition effects whenever a transform spans data from multiple prediction blocks. Moreover, this filtering technique enables extending the flexibility in decoupling prediction and transform structures, as various restrictions may now be eliminated. In particular, it makes it possible and beneficial to allow a transform block to span residual data from both inter and intra predicted blocks, whereas HEVC necessarily forces a single type of prediction in each coding unit. The method is further extended to include motion refinement that accounts for the pre-filtering approach. Experiments provide evidence for consistent coding gains over HEVC and VP9.

Index Terms— Transform coding, HEVC, VP9, video coding

1. INTRODUCTION

Transform coding is a central component in video compression tasked with reducing the inherent spatial redundancy in the prediction residual. Current mainstream video codecs apply the transform to a block of prediction residual obtained through either motion-compensated (temporal) prediction, or intra (spatial) prediction. Traditionally, the transform block has been constrained to fall within the boundaries of a prediction block (PB) (see for example VP9 and H.264 [1, 2]). Recently, more flexibility has been considered in terms of some decoupling of the transform block partition from the prediction block partition. For example, HEVC allows a transform block (under certain conditions) to span multiple PBs [3]. This flexibility has the potential of enhancing coding efficiency by exploiting correlations across a larger transform block, especially in areas with relatively flat prediction residuals. Fig.1 depicts two different partitions into transform blocks, and in this example, assuming all blocks are inter-predicted, HEVC would allow both transform unit (TU) structures shown, while earlier codecs would only allow the one shown on the right due to restrictive coupling with the prediction unit (PU) structure.

This work was supported by Google, Inc.

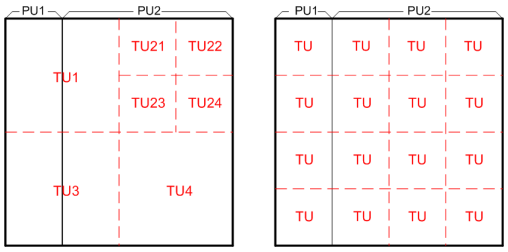

Fig. 1. An example of prediction and transform structures in HEVC. nLx2N PU structure is used.

However, decoupling the transform structure from the prediction structure comes at the risk of side effects due to potential discontinuity in prediction, which may severely impact the performance of the transform. For instance, TU1 and TU3 in Fig.1(L) could be disturbed by blocking artifacts around the edge separating PU1 and PU2. Such artifacts typically arise in conjunction with PUs that are independently motion compensated because: (i) motion search only based on block matching might assign very different motion vectors of neighboring blocks, unlike the typically smooth changes in the actual optical flow; and (ii) real motion vectors do not always reflect pure translation which is well represented by current 2-D motion models. This phenomenon motivated research on means to mitigate the discontinuity produced by over-simplified motion compensation models. Some recent efforts focused on combining intra and inter prediction, so as to handle blocks that are not well captured by either option. See e.g. joint inter-intra predictors based on either fixed averaging weights[4] or more adaptive joint models[5, 6], as well as our own related recent work, motivated by the need to capture situations where a prediction block overlaps both a portion of a moving object and some background, and where a family of geometric partitions of the block allow subdividing it irregularly into an intra and an inter part[7]. Earlier efforts addressed the problem at the broader motion description level and led to approaches that capture more complex motion than is afforded by simple block translation. A control grid interpolation of motion compensation was proposed to perform spatial image transformations by interpolating the per-pixel motion vector using motions determined for several anchor points in the image[8]. Another hierarchical motion model developed in [9] considers linear and affine motion flows in addition to pure translation. Such prior work employs sophisticated motion models and dramatically increases the codec complexity in motion model estimation as well as in motion interpolation. In contradistinction, without making massive changes to current motion estimation algorithms, we first propose a pre-filtering technique which combines block match-

ing results for multiple PUs by edge-directed filters, hence smoothing boundary transitions and enhancing the efficiency of subsequent transform coding.

It is important to note that existing codecs that allow some prediction-transform structure decoupling, such as HEVC, do so in a very restrictive fashion due to several constraints that must be imposed on the underlying PUs in the coding unit (CU), namely, it is required that they all employ the same type of prediction (either inter or intra). The reasons include: mixing inter and intra PUs in the same TU would cause major boundary transition effects; and intra predicted blocks require access to some reconstructed pixels from adjacent blocks; both of which severely limit the flexibility of the quadrant tree based TU structure when applied across compound residual blocks, and led to the above mentioned constraints. In this paper, given the proposed pre-filtering approach's ability to substantially mitigate the boundary transition effects, we further extend the flexibility of prediction-transform structure decoupling to such compound coding unit blocks, to the full extent possible given the directionality of the intra PUs involved. The overall enhanced flexibility and boundary transition effect mitigation yield consistent coding gains as will be evidenced in the results section. Specifically, It is experimentally demonstrated that the methods reduce the bit-rate consistently over a set of video sequences at various resolutions. Moreover, at low bit rates, where much of the residual is quantized to zero, the pre-filtering technique has an important perceptual role as it helps preserve necessary continuity and reduces visual artifacts due to artificial discontinuities.

2. PRE-FILTERING TECHNIQUE FOR TRANSFORM CODING ACROSS PREDICTION BLOCK BOUNDARY

Consider a transform block overlapping two inter prediction blocks. To simplify the presentation and implementation, we specify the partition by the function of the boundary line

$$f(x,y) = a_1(x - a_2\frac{w}{4}) + a_3(y - a_4\frac{h}{4}) = 0, \qquad (1)$$

for the $w \times h$ coding unit block, where $(a_2\frac{w}{4}, a_4\frac{h}{4})$ denotes the coordinates of a pixel on the line and the partitioning direction is determined by a_1 and a_3. In the example of Fig.1, the corresponding parameters (a_1, a_2, a_3, a_4) are $(1, 1, 0, 0)$. Suppose we use the two motion vectors obtained for the respective prediction blocks PU1 and PU2, to generate two motion-compensated versions of the entire coding unit, and denote them by $p_1(x,y)$ and $p_2(x,y)$, respectively. The conventional motion compensated prediction of the entire coding unit block can be stated as

$$p(x,y) = w_1(x,y)p_1(x,y) + w_2(x,y)p_2(x,y) \qquad (2)$$

with binary weights trivially given by

$$w_1(x,y) = \begin{cases} 0, & \text{if } f(x,y) \geq 0 \\ 1, & \text{if } f(x,y) < 0, \end{cases}$$

$$w_2(x,y) = \begin{cases} 1, & \text{if } f(x,y) \geq 0 \\ 0, & \text{if } f(x,y) < 0. \end{cases} \qquad (3)$$

As we observed in our experiments, predicting PUs independently, especially when different reference frames are used, will often result in spurious high frequency components due to the boundary. The introduction of such high frequency components compromises the energy compaction of the transform, and the compression efficiency. The natural way to avoid this problem is to replace the binary weights of (3), with a softer sequence of weights.

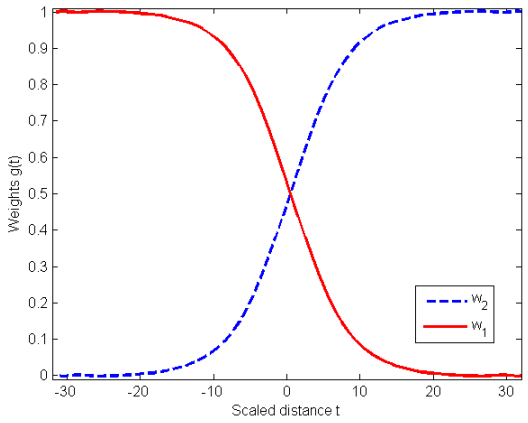

Fig. 2. The weighting function applied to reduce discontinuity between predictions in different PUs.

2.1. Filter Design

A 1-D "soft step" weighting function $g(t)$ (see Fig.2) is designed to combine two 1-D predictions smoothly close to the origin while approaching binary weights further away from it. To generate the needed 2-D predictions, the above 1-D function is applied along the direction perpendicular to the boundary line. Specifically, the two extended predictors for each pixel (x,y) in the coding unit are mixed as described in (2) with weights $w_1(x,y)$ and $w_2(x,y) = 1 - w_1(x,y)$, where $w_1(x,y)$ is obtained as:

$$w_1(x,y) = g\left(\frac{f(x,y)}{\sqrt{a_1^2 + a_3^2}}\right) \qquad (4)$$

where the argument of $g(\cdot)$ measures the distance from the boundary. Combining two extended block predictions that both likely contain true textures within the extended region, this soft binary weighting scheme differs from deblocking filters that indiscriminately smooth out textures. Thus we expect to efficiently recoup the benefits of large and flexible block transform without losing much accuracy in prediction.

2.2. Motion Refinement for Pre-filtered Inter Prediction

Motivated by the recognition that motion vectors independently optimized for PU1 and PU2 are not necessarily optimal for the filtered prediction, we propose a final step to refine the initial motion vectors obtained by minimizing errors (SAD or HAD) of non-filtered predictions, while accounting now for the joint optimality of these motions given that the final prediction is obtained as a weighted combination.

Let $p_o(x,y)$ and $p_{\cdot}^{mi}(x,y)$ denote the original pixel value and the motion compensated pixel determined by motion information mi including reference frame(s) and motion vector(s). Given the weighting filters w_1 and w_2 determined by the PU partition, we initialize motion information mi_1 and mi_2 to the values obtained by separate motion estimation for PU1 and PU2 (generated by the standard encoding process) and refine them as follows:

1. Calculate $p_{wo1}(x,y) = p_o(x,y) - w_2(x,y)p_2^{mi_2}(x,y)$.
2. Run a weighted motion search for the whole CU to determine mi_1^* minimizing

$$\sum |p_{wo1}(x,y) - w_1(x,y)p_1^{mi_1^*}(x,y)|.$$

3. Update mi_1 as mi_1^*.
4. Calculate $p_{wo2}(x,y) = p_o(x,y) - w_1(x,y)p_1^{mi_1}(x,y)$.
5. Run a weighted motion search for the whole CU to determine mi_2^* minimizing
$$\sum |p_{wo2}(x,y) - w_2(x,y)p_2^{mi_2^*}(x,y)|.$$
6. Update mi_2 as mi_2^*.

The SAD of filtered prediction for the whole CU monotonically decreases in update steps 3 and 6. While the above process can be iterated multiple times to further improve the filtered prediction, we only run it once in the experiments to avoid unnecessary computational cost.

Note that the pre-filtering technique can also be generalized to the other PU partitioning method which split a CU into four quadrant PUs. In our preliminary implementation, we retained the coding method for this partition unchanged from HEVC while noting that an extension of the above algorithm that would require more space will be developed and is hence beyond the scope of this paper.

3. PRE-FILTERING FOR ENCODING A COMPOUND CU

Based on our earlier discussion in Sec.1, we introduce a new type of CU, called compound CU, which contains both intra PU(s) and inter PU(s). It is introduced in order to enhance standard prediction/transform structures and allow better prediction. Recall that to retain the flexibility of TU structures, the intra PU should not depend on other PU(s) within the same CU, and this can be ensured by only allowing PU structures for a compound CU as illustrated in Fig.3, where the only intra PU is located in the top or the left corner.

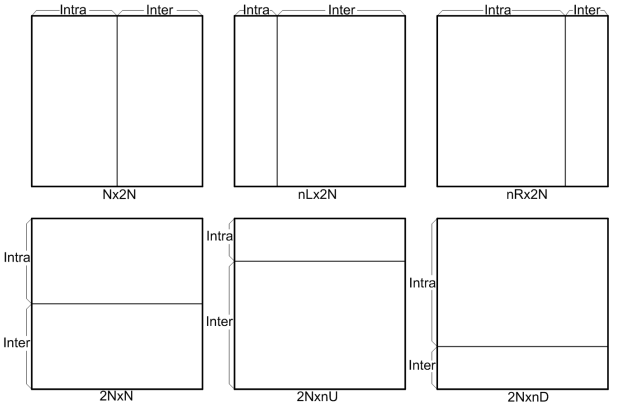

Fig. 3. PU structures for proposed compound coding unit

3.1. Pre-filtering compound CU

An obvious concern with compound CU is that the boundary between intra and inter residual blocks will cause more severe effects than the transition between inter blocks. The discontinuity is exacerbated due to the different nature of intra and inter residuals. Intra residuals tend to have energy that increases with distance from available boundaries, while inter residuals tend to be distributed independent of position in the block. This may explain why HEVC does not allow compound CUs. However, the pre-filtering technique in Sec.2 opens the door to allowing and exploiting compound CUs. Specifically the intra and inter predictions, extended to the entire CU are combined according to

$$w_{intra}(x,y)p_{intra}(x,y) + w_{inter}(x,y)p_{inter}(x,y) \qquad (5)$$

where the coefficients $w_{intra}(x,y)$ and $w_{inter}(x,y)$ are determined exactly as done for $w_1(x,y)$ and $w_2(x,y)$ in Sec.2.

3.2. Intra mode and inter motion refinement

Similar to the consideration in Sec.2.2, the optimal prediction for a compound CU is obtained by a joint search over the intra mode k for the intra PU and the motion information mi for the inter PU. Both k and mi are initialized by values obtained separately for the corresponding PUs via the standard encoder decisions. To achieve joint optimality for pre-filtered compound CU, subsequent refinement of k and mi is executed by the algorithm below:

1. Calculate target weighted intra prediction
$p_{wointra}(x,y) = p_o(x,y) - w_{inter}(x,y)p_{inter}^{mi}(x,y)$.
2. Run a weighted intra mode search for the whole CU to determine intra mode k^* minimizing
$\sum |p_{wointra}(x,y) - w_{intra}(x,y)p_{intra}^{k*}(x,y)|$.
3. Update k as k^*.
4. Calculate target weighted inter prediction
$p_{wointer}(x,y) = p_o(x,y) - w_{intra}(x,y)p_{intra}^{k}(x,y)$.
5. Run a weighted motion search for the whole CU to determine mi^* minimizing
$\sum |p_{wointer}(x,y) - w_{inter}(x,y)p_{inter}^{mi^*}(x,y)|$.
6. Update mi as mi^*.

Here $p_{intra}^k(x,y)$ denotes the intra prediction created by intra mode k. The prediction error in the SAD metric for the compound CU is also guaranteed to monotonically decrease by steps 3 and 6 and here to we only perform one iteration in experiments to minimize complexity.

4. EXPERIMENTAL RESULTS

We first quantitatively evaluate the two proposed methods by modifying the HEVC reference software. The pre-filtering technique is added as a new coding option to inter-coded CUs, where the TU structure includes transform blocks that span over multiple PUs. The codec is referred to as Coder A. The compound inter/intra CU is then further enabled as a second additional mode on top of Coder A, and is referred to as Coder B. The pre-filtering process is enforced for compound coded CUs.

The experiments were based on the random access (main) configuration with QP values ranging from 22 to 37. Video clips with resolutions from CIF to 720p were coded with GOP = 8 and intra frame interval equivalent to 32. The performance gains, in terms of BD rate reduction, for several sequences were presented in Table 1. Clearly, both techniques provided consistent gains over the reference software. The proposed techniques were also tested in the experimental branch of the VP9 framework, where a preliminary implementation for transform spanning multiple pre-filtered prediction blocks was enabled for inter blocks that were generated from a single reference frame. Consistent performance improvements were obtained as shown in Table 2.

Moreover, a perceptual coding quality comparison at low bit-rate is presented in Fig.4. Apparently the proposed pre-filtering approach achieves better perceptual quality around the contour of legs due to reduced blockiness in prediction.

Table 1. BD rate reduction due to the proposed approaches relative to the HEVC reference software. Coder A: HEVC with pre-filtering technique applied to inter CUs, Coder B: HEVC with pre-filtering technique applied to both inter CUs and compound CUs.

Resolution	Sequence	Coder A	Coder B
CIF	Akiyo	0.4919	0.8918
	Hall Monitor	1.0742	1.3352
	Silent	0.4986	1.1457
832×480	BQMall	1.4039	1.8986
	Party Scene	0.9839	1.9184
720p	City	1.0324	1.6910
	Mobile Calendar	1.4381	2.1349

Table 2. BD rate reduction due to the proposed approaches relative to the VP9 reference software. Coder A: VP9 with pre-filtering technique applied to inter residuals, Coder B: VP9 with pre-filtering technique applied to both inter and compound residual blocks.

Resolution	Sequence	Coder A	Coder B
CIF	Akiyo	0.3857	0.4789
832×480	BQMall	0.9687	1.2031
720p	City	0.4233	0.8280

Fig. 4. Visual comparison: a patch from frame 82 in sequence $BQMall$. (L) Uncompressed; (M) Coded by HEVC reference software; (R) Coded by the proposed methods.

5. CONCLUSION

A pre-filtering technique employed across multiple prediction blocks is proposed to fully realize the potential benefits of decoupling the transform block dimension from the prediction block partition. The pre-filtering mitigates the potential discontinuity artifacts at prediction block boundary that often severely compromise the transform coding efficiency. This filtering technique also enables extending the flexibility in decoupling transform-prediction structures by eliminating various restrictions that had been necessary until now, including in particular the use of compound inter/intra coding units, where the transform block overlaps both inter and intra prediction blocks. Experimental results demonstrate that the pre-filtering approach, in conjunction with the introduction of compound coding units, provides consistent coding gains for both HEVC and VP9 codecs.

6. REFERENCES

[1] J. Bankoski, R.S. Bultje, A. Grange, Q. Gu, J. Han, J. Koleszar, D. Mukherjee, P. Wilkins, and Y. Xu, "Towards a next generation open-source video codec," *IS&T/SPIE Electronic Imaging*, 2013.

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, 2003.

[3] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, 2012.

[4] J. Xin, K. N. Ngan, and G. Zhu, "Combined inter-intra prediction for high definition video coding," *Picture Coding Symposium*, 2007.

[5] Y. Chen, J. Han, T. Nanjundaswamy, and K. Rose, "A joint spatio-temporal filtering approach to efficient prediction in video compression," *Picture Coding Symposium*, 2013.

[6] J. Seiler, T. Richter, and A. Kaup, "Spatio-temporal prediction in video coding by non-local means refined motion compensation," *Picture Coding Symposium*, pp. 318–321, 2010.

[7] Y. Chen, D. Mukherjee, J. Han, and K. Rose, "Joint inter-intra prediction based on mode-variant and edge-directed weighting approaches in video coding," *to appear in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[8] G. J. Sullivan and R. Baker, "Motion compensation for video compression using control grid interpolation," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2713–2716, 1991.

[9] R. Mathew and D.S. Taubman, "Quad-tree motion modeling with leaf merging," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 1331–1345, 2010.