VARIABLE BLOCK-SIZE OVERLAPPED BLOCK MOTION COMPENSATION IN THE NEXT GENERATION OPEN-SOURCE VIDEO CODEC

Yue Chen, Debargha Mukherjee

Google Inc., Mountain View, USA
Email: {yuec, debargha}@google.com

ABSTRACT

Traditional motion compensation is based on a simplistic assumption of a piecewise constant and pure translational motion field. Premised on the inefficiency of such block-copying-based algorithms in compressing real-life videos, this paper proposes overlapped block motion compensation for AV1, a next generation open-source video codec. Motions assigned to surrounding blocks will contribute to predicting a current block, via a well-defined overlapping scheme appropriately designed for advanced variable block-size partitioning frameworks. To efficiently incorporate the proposed tool, the encoder is optimized with proper early termination to manage complexity in conjunction with weighted motion search accounting for non-uniformly weighted distortion in overlapped prediction. Moreover as an extension, by re-arranging decoding operations, we enable non-causal overlapping which also utilizes motions of bottom and right neighbors. Being verified by integration in the AV1 codec, the tools achieve considerable coding gains on various test sets.

Index Terms— Motion compensation, inter prediction, OBMC, video compression, AV1

1. INTRODUCTION

Motion compensation techniques largely contribute to the success of modern video compression tools by effectively reducing the temporal redundancy in video signals. In mainstream video codecs, e.g. AV1[1], VP9[2], HEVC[3], and h.264[4], it is built upon block matching. Block matching is efficient at handling uniform translational motion when being complimented by features like adaptive sub-pixel interpolation and motion vector referencing. But it still often under-represents the temporal relationship between frames in natural video signals because of the complicated motion of camera and objects as well as irregularly shaped moving objects that cannot be perfectly segmented.

Although many computer vision algorithms can provide good analysis of motion and segmentation, in the scenario of video compression, either the bit cost to convey the accurate models or the computation spent on model estimation still holds back the adoption of these techniques. To overcome the over-simplification of block based single reference prediction without reinventing the fundamental of temporal prediction, multi-hypothesis motion compensation algorithms were proposed to superimpose multiple motion compensated prediction, for example, bi-directional compound prediction, compound prediction using more than two references[5] and overlapped block motion compensation (OBMC)[6, 7]. Overlapped block prediction, proposed and implemented back in the era of h.263, has been proved to largely reduce prediction errors. The basic idea is, assuming the frame is partitioned into blocks of the same size and each is assigned its own motion vector, OBMC creates a smoothly combined prediction between the centers of 4 adjacent blocks by applying bell shaped masks peaked at each block center to each extended motion compensated block and adding up the weighted predictions. However, due to the complications introduced by variable block size partitioning and mixed inter-intra coding, this technique has not been extensively implemented in recent literature. In [8], a frame using variable block-sizes was virtually partitioned into the smallest block size to make the standard OBMC applicable to general cases. In other work, the filter coefficients for different predictors at each pixel location are made inversely proportional to the distances to the center of the original prediction units. Based on it, [9] combined the current predictor with one nearest predictor adjacent to the top or left edge to implement causal OBMC, and [10] supported blending with two or more neighbors next to the four edges. Such distance-modulated weighting is not very friendly to SIMD optimization, because the distances and weights need to be computed on-the-fly pixel by pixel and usually the weights turn out to be irregularly distributed. In [11], OBMC is implemented in the Daala codec in a partitioning framework that is restrained to work with a recursive weights computation. It is worth mentioning that most of the prior work, especially for 4-sided OBMC [6, 7, 8, 10], applies OBMC to the entire frame assuming all blocks are inter coded, therefore limits the flexibility of the codec.

In this paper, we first revisit the causal OBMC technique, and propose a practical weighting mechanism based on two-stage 1-D filtering which can easily fit in advanced partitioning frameworks. In an extended experiment, to maximize the benefit of overlapped prediction by using motion vectors assigned to future blocks, the coding process is re-designed to resolve the non-causal referencing in non-causal OBMC (NCOBMC). These tools are added as extra options for inter prediction, competing with other existing tools in a rate-distortion optimization loop assisted by an effective early termination to achieve manageable encoding complexity. The overlapped prediction is also optimized by weighted motion estimation, which is exactly designed in accordance with filtering in OBMC.

2. VARIABLE BLOCK-SIZE OBMC

Advanced codecs exploit variable block-size partitioning to fit various coding tools into rectangular regions of proper sizes. The most common method is the recursive quad-tree decomposition[1, 2, 3], a versatile framework capable of directly operating in static and large areas to save bitstream overheads or going down to small corners to match the shapes of objects. However, on the other hand, such flexibility introduces a significantly increased number of possible partitioning patterns for the surrounding area of a block, making the inter-block interactions, like combining prediction blocks via 2-D masks, especially between a number (more than one above and one

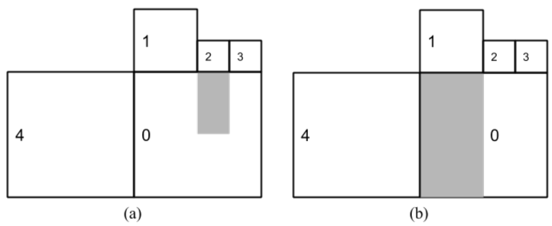

Fig. 1. Overlapping regions

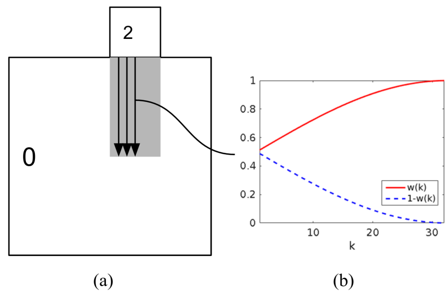

Fig. 2. Weighting based on 1-D raised cosine window

on the left) of coded neighbors and the current block, very difficult to define and hard-code in the codec.

Therefore, in our OBMC framework, to bypass the complication of directly defining 2-D masking for arbitrarily partitioned neighborhood, we propose a progressive 1-D overlapping algorithm. It is operated in the horizontal and the vertical directions separately using 1-D filters, with each surrounding predictor working in a restricted region that does not heavily tangle with others'.

2.1. Causal OBMC for Variable Block-size Coding

This algorithm starts with initializing the overlapped prediction $p_{obmc}(x, y)$ as the basic prediction by applying the motion assigned to the current block. For an $N \times M$ prediction unit, let $p_i(x, y)$ ($0 \leq x < M, 0 \leq y < N$) denote a prediction block for pixels in the current block ($i = 0$) but computed using the reference and motion vector combination(s) of block i, so $p_{obmc}(x, y)$ is initialized as $p_0(x, y)$. Note that both compound and single reference frame predictors are allowed to contribute to OBMC, so p_i could be a combined prediction, which averages two single-reference predictions in the AV1 codec.

Next, prediction overlapping will be performed progressively on top of the continuously updated $p_{obmc}(x, y)$ in two stages: first with the adjacent blocks above, and next with those to the left. For example, in Fig.1, block 0 is the unit to predict, block 1-4 are coded adjacent blocks. At the first stage, we check the predictors of block 1-3, located right on top of block 0. Among them, if block i is coded by an inter predictor, the assigned motion parameters are exploited to compute a new prediction block $p_i(x, y)$ for the 'overlapping region' of p_0 and p_i in block 0: a rectangular area in block 0, with the common edge of block 0 and block i as one of its edges and the opposing edge on the mid-line of block 0. For instance, in Fig.1(a), the shaded area is the overlapping region of p_0 and p_2 for pixels in block 0, and the predictor blending will only be operated there. A K-tap 1-D filter $w_K(k)$, which is shaped as a half raised cosine window

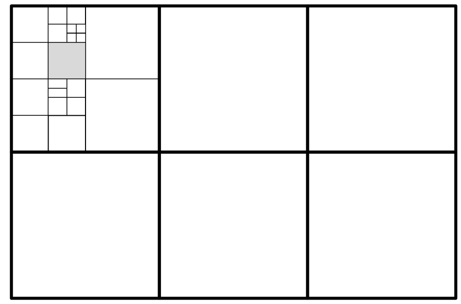

Fig. 3. Grouping prediction blocks to implement non-causal OBMC

(see Fig.2(b) for a 32-tap example) and formulated as

$$w_K(k) = \frac{1}{2}sin(\frac{\pi}{2K}(k + \frac{1}{2})) + \frac{1}{2}, k = 0, 1, ..., K - 1 \quad (1)$$

is applied to every column (see Fig.2(a)) of the overlapping region, and updates $p_{obmc}(x, y)$ as

$$w_{\frac{N}{2}}(y)p_{obmc}(x, y) + (1 - w_{\frac{N}{2}}(y))p_i(x, y). \quad (2)$$

This filter approximately averages the predictions at the common edge, and gradually reduces the influence of the new prediction p_i until it vanishes at the mid-line of the current block, because the conventional block matching p_0 often works best for pixels in the center. Then we move on to the second stage to exploit predictors of the left neighbors. Likewise, 1-D filtering will be performed on top of the $p_{obmc}(x, y)$ updated after the first phase: (1) the overlapping region for each left neighbors will be on the right side of the common edge, e.g. the shaded area for p_4 in Fig.1(b); (2) we apply the 1-D filter in the horizontal direction, i.e.

$$p_{obmc}(x, y) := w_{\frac{M}{2}}(x)p_{obmc}(x, y) + (1 - w_{\frac{M}{2}}(x))p_i(x, y). \quad (3)$$

Note that the new predictors at the second stage could share part of their overlapping regions with those at the first stage, e.g. block 1 and block 4 in Fig.1, in such cases, it means that the new prediction will be processed on top of a combined prediction rather than on the original p_0, so the final prediction for the top-left quarter of the current block could be the combination of three predictors. Therefore the two-stage 1-D filtering based on one fixed filter in (1) will slightly bias the final prediction towards the left predictors compared to how much we dependent on the above predictors. We still determine to use this simple scheme, because none of the experimented fixes gives consistent additional gain, including avoiding two way overlapping or switching to 2-D windows as in [7] when combining four predictors (only if the current, top, top-left, and left blocks are inter blocks of the same size). As a future work, coefficient optimizations for each stage and even for different dimensions of overlapping will be explored to enhance the capability of this framework.

2.2. Non-causal OBMC for Variable Block-size Coding

Non-causal overlapped prediction is also implemented. Assuming all neighbors' motion parameters are accessible, this extension can easily be realized by a similar way as in Sec.2.1 to cooperate with neighbors in four stages, at each with bottom, right, top and left predictors. Therefore, the only issue left to us is the accessibility of parameters of bottom and right neighbors. In current codecs, at the time of prediction, parameters assigned to 'future' units are not reachable, because parameter decoding, prediction and reconstruction are done back-to-back for each prediction unit.

Therefore we re-arrange the pipeline of the decoder to obtain access to motion parameters at a 'future' position, without substantially harming the throughput. We segment the frame into 64×64 chunks, illustrated by the bold lines In Fig.3, which consist of prediction blocks. Within each chunk, communication of prediction parameters and the final predictive coding are completely separated into two phases for the whole group, i.e. all their parameters are decoded together at one time before we work on prediction and reconstruction. This makes motion parameters within the same group being 'randomly' accessible at the time of prediction, since parameters of 'future' blocks in the same chunk have been decoded and buffered in advance. In our experiment, non-causal OBMC is enabled for blocks with bottom or right neighbors located in the same chunk, while across the chunk borders it degrades to partly non-causal(3-sided) or causal(2-sided) OBMC.

3. ENCODER OPTIMIZATION

3.1. Bitstream Syntax

State-of-the-art compression softwares have a lot of tools to enhance the rudimentary concept of temporal block copying, including but not limited to compound prediction and sub-pixel motion using switchable interpolation filters, therefore overlapped block prediction does not always provide a substantial advantage over the existing methods. In the two experimental codecs that we implement, one with causal OBMC and the other with non-causal OBMC, the proposed algorithms are integrated as a second option for inter blocks $\geq 8\times 8$ in addition to non-overlapped prediction. In the bitstream, right after the motion parameters being coded, one binary symbol per block is indicated to invoke OBMC. At the decoder side, non-overlapped inter prediction is first computed using the assigned motion parameters, and then if OBMC mode is signaled, the overlapping algorithms in Sec.2 are applied. Based on this operation logic, in this section, we discuss how the encoder efficiently assigns a good combo of prediction parameters (reference frames, motion vectors, the OBMC flag, and etc.) to each prediction unit.

3.2. Fast RDO for Causal OBMC

Ideally, to figure out the best coding tool for each block, all possible combinations of prediction options should be fully tested. In particular for this experiment, it means prohibitively expensive test encodings for both non-overlapped and overlapped prediction for all possible reference frame and motion vector combinations. In the proposed framework, non-overlapped predictors are tested as the baseline AV1 codec does. While for OBMC predictors, instead of seeking the optimality, we omit those built upon low-quality base predictors, which result in non-overlapped prediction block with a luma RD cost higher than the current best total RD cost (for both luma and chroma planes) among the tested motion parameters. By performing this early termination, we reduce the extra encoding time by over half, while only compromise the coding gain by 0.07% on our CIF test clips.

3.3. Motion Estimation for Causal OBMC

For causal overlapped block prediction, the conventional block matching is not sufficient to account for the non-uniformly weighted inaccuracy at different pixel locations. Therefore, when testing a motion vector mode that allows a new vector being differentially coded, a special motion estimation is executed based on a weighted distortion metric. The new metric considers the actual per-pixel distortion of the overlapped prediction. First of all, the final overlapped prediction can be generally formulated as a linear combination of all the predictors using 2-D filters, i.e.

$$p_{obmc}(x,y) = \sum_i f_i(x,y)p_i(x,y). \tag{4}$$

Given the uncompressed pixel value $s(x,y)$, the prediction error is

$$d(x,y) = s(x,y) - \sum_i f_i(x,y)p_i(x,y). \tag{5}$$

Next, we sort out the variables that are known before motion estimation for the current block. Those include the source video $s(x,y)$, predictions computed from surrounding predictors $p_i(x,y)(i \neq 0)$, and the final 2-D weights $f(x,y)$ since they are only dependent on partition and the inter/intra selections of neighbors. So the formula can be re-arranged as

$$d(x,y) = s(x,y) - \sum_{i\neq 0} f_i(x,y)p_i(x,y) - f_0(x,y)p_0(x,y), \tag{6}$$

where the result of the first two terms is actually our target weighted prediction

$$t(x,y) = s(x,y) - \sum_{i\neq 0} f_i(x,y)p_i(x,y). \tag{7}$$

In our experiment, no matter the distortion to consider is SAD, SSE or variance, motion search for OBMC will be based on weighted pixel differences,

$$d(x,y) = t(x,y) - f_0(x,y)p_0(x,y), \tag{8}$$

with $t(x,y)$ and $f_0(x,y)$ pre-calculated at one time before testing candidate predictors. Although we cannot represent $t(x,y)$ and $f_0(x,y)$ in simple closed-form expressions, they can be computed by walking through the 2-stage overlapping process described in Sec.2.1. Specifically, $\sum_{i\neq 0} f_i(x,y)p_i(x,y)$ in $t(x,y)$ is equivalent to $p_{obmc}(x,y)$ assuming p_0 is all-zero, and likewise $f_0(x,y)$ is the result of faking all-zero $p_i(i \neq 0)$ and all-one p_0. In reality, it is only enabled for causal OBMC when the current predictor uses one reference frame, because the weighted joint motion search for two vectors turns out to offer a very expensive gain-complexity trade-off. And for non-causal OBMC, more work is required to resolve the motion vector dependency of future blocks on the current vector.

3.4. Two-pass RDO for Non-causal OBMC

In the codec supporting non-causal OBMC, the encoder has to determine both motion parameters and NCOBMC flag. If only performing one pass RDO on each block, this extra flag cannot be decided because there is no way to compute NCOBMC prediction in the absence of future neighbors' motions. Therefore, a two-pass RDO is proposed. In the first pass, for each 64x64 chunk, all of the parameters, including partition and motion, are picked assuming OBMC is causal, in other words, using the RDO described by Sec.3.2 and Sec.3.3. Afterwards, with partitions and motion parameters fixed, in the same causal scan order, a second pass RDO visits each inter predicted unit in the chunk to finalize the NCOBMC flag based on the RD costs of regular inter prediction and NCOBMC prediction. Note that this implementation of NCOBMC is still preliminary and has much space for improvement. For example, NCOBMC can be tested with other candidate motions besides the one picked by the first pass.

4. EXPERIMENTAL RESULTS

The proposed overlapped prediction algorithms have been integrated in AV1[1], the open-source and royalty-free video codec developed by Alliance of Open Media. The codec is available in the online repository, causal obmc mode will be enabled using the configuration: *–enable-experimental –enable-motion_var*, or the codec will switch to incorporating non-causal obmc if a third configuration *–enable-ncobmc* is added. The codec runs in the high-latency setting (most frames are B frames) with most speed-up features turned off (*–cpu-used=0*). To make a comprehensive study, we encode 150 frames of each test clip at a wide range of target bitrates.

Using the PSNR metric, we compare three codecs, including (1) the AV1 baseline itself, (2) OBMC: with causal obmc added as an extra coding tool, and (3) NCOBMC: with non-causal obmc enabled (causal obmc is disabled in the third codec). On three test sets, lowres (40 clips of 240p, SIF, or CIF resolution), midres (30 clips of 4cif, 480p, or 832×480 resolution), and hdres (19 720p clips, 19 1080p clips, and 1 xga clip), the average difference is presented in BDRate[12] in Table 1. Since a negative BDRate means better compression efficiency, both the proposed tools consistently save the bit-rate by 2.2-2.8% for different resolution, and the codec with non-causal overlapped prediction mode, although not fully optimized yet, outperforms the codec with causal OBMC by about 0.3-0.4%.

Table 1. BDRate(%) of the proposed experiments for test sets with different resolution, in comparison with the AV1 baseline

	lowres	midres	hdres
OBMC	-2.325	-2.241	-2.204
NCOBMC	-2.845	-2.638	-2.538

Table 2. BDRate(%) of the proposed experiments for individual clips, in comparison with the AV1 baseline

Resolution	Sequence	OBMC	NCOBMC
240p	basketballpass	-2.513	-2.865
	bqsquare	-3.559	-4.773
CIF	akiyo	-2.868	-3.544
	bowing	-2.039	-2.041
	waterfall	-4.605	-5.813
	soccer	-1.922	-2.143
480p	BQMall	-3.456	-4.637
	Mobisode2	-1.738	-1.739
	RaceHorses	-2.714	-3.281
	ducks_take_off	-2.034	-2.186
720p	city	-2.568	-3.466
	cyclists	-2.023	-2.150
	factory	-2.735	-3.694
	vidyo4	-2.311	-2.883
1080p	riverbed	-0.418	-0.356
	shields	-2.599	-3.414
	parkscene	-1.999	-2.134
	station2	-4.862	-5.771

The BDRates on selected individual sequences are also listed in Table 2, where we can see the proposed tools improves the compression for all the clips (also for clips in the test sets but not listed here), and especially work well on clips with low to moderate motion that are hard to fully compensate by block matching. For example, *factory* is an animation with various rotating and warping objects, *city* is shot from changing viewing angle, and *basketballpass* has both camera motion and running athletes. Even for clips that looks very static, e.g. *akiyo* and *bowing*, overlapped prediction shows an advantage in treating the tiny non-translational motions at object edges. Our investigation also further goes down to the block level, the frequency of causal OBMC being chosen by blocks of different sizes is presented in Table 3. Interestingly, compared to square blocks, non-square blocks have a strong preference for OBMC. A reasonable intuition is that the codec ends up with those partitions usually for handling off-grid or irregular object edges, where uniform prediction is apparently not sufficient. For square blocks, CIF clips tend to apply OBMC to smaller blocks more often, while the statistics become more stationary on 1080p clips possibly because of flatter motion variation. And for both resolution, as presented by Avg2 in Table 3, OBMC-predicted region takes about 30% of inter frames.

Table 3. Percentage(%) of causal OBMC usage (Avg1: based on counts; Avg2: based on area.)

	CIF	1080p		CIF	1080p
8×8	73	28	32×32	4	32
8×16	84	31	32×64	83	78
16×8	77	87	64x32	51	81
16×16	19	27	64x64	6	8
16×32	85	75	Avg1	58	44
32×16	59	76	Avg2	33	31

Table 4. Additional coding time (%) of the codec supporting causal OBMC, in comparison with the AV1 baseline

	Encoding	Decoding
CIF	17	5
1080p	30	9

We also test the complexity of the causal OBMC codec on CIF and 1080p clips at mediate bitrates. The additional computations, in comparison with the time consumed by the baseline codec, are listed in Table 4, demonstrating low decoding complexity overhead as well as manageable encoding complexity. Also, it is worth mentioning that if removing the weighted motion search and reusing the vector found by block matching, we will sacrifice around 0.4% performance gain in exchange for 5% and 12% less extra encoding time on CIF and 1080p sequences.

5. CONCLUSION

In this paper, we introduce a novel and practical framework to exploit overlapped block motion compensation in advanced codecs. The proposed approach also involves an efficient mode decision module and a motion search algorithm matching the objective of OBMC. By modifying the decoding pipeline, a preliminary extension to non-causal OBMC is also implemented and demonstrates promising gains. The proposed tool achieves consistent compression improvement at the cost of reasonable extra coding complexities. Future work includes replacing off-the-shelf windows by fully optimized 1-D filters and designing RDO that realizes the true potential of non-causal OBMC.

6. REFERENCES

[1] "Alliance for open media," http://aomedia.org.

[2] 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.

[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] 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.

[5] M. Flierl, T. Wiegand, and B. Girod, "Video codec incorporating block-based multihypothesis motion-compensated prediction," *Proc. SPIE 4067 Visual Communications and Image Processing*, May 2000.

[6] H. Watanabe and S. Singhal, "Windowed motion compensation," *Proc. SPIE Conf. Visual Commun. Image*, vol. 1605, pp. 582–589, 1991.

[7] M. T. Orchard and G. J. Sullivan, "Overlapped block motion compensation: An estimation-theoretic approach," *IEEE Transactions on Image Processing*, vol. 3, no. 5, 1994.

[8] J. Zhang, M. O. Ahmad, and M. N. S. Swamy, "Overlapped variable size block motion compensation," *Proceedings of International Conference on Image Processing*, vol. 3, pp. 642–645, 1997.

[9] Y. Chen, T. Wang, Y. Tseng, W. Peng, and S. Lee, "A parametric window design for OBMC with variable block size motion estimates," *IEEE International Workshop on Multimedia Signal Processing*, pp. 1–5, 2009.

[10] S. Park, S. H. Kim, C. Kim, and S. Lee, "Adaptive windowing technique for variable block-size motion compensation," *Proc. SPIE 5308 Visual Communications and Image Processing*, 2004.

[11] T. B. Terryberry, "Adaptive motion compensation without blocking artifacts," *https://people.xiph.org/tterribe/daala/vbs obmc.pdf.*

[12] G. Bjontegaard, "Improvements of the BD-PSNR model," *VCEG-AI11 ITU-T SG16 Q.6 Document*, July 2008.