Improvements on Intra Block Copy in Natural Content Video Coding

Haoming Chen*, Yu-Sheng Chen*, Ming-Ting Sun*
Ankur Saxena**, Madhukar Budagavi**
* University of Washington, Seattle, WA 98105 USA
{eehmchen, yschen}@uw.edu, sun@ee.washington.edu
** Samsung Research America, Richardson, TX 75082, USA
{a.saxena1, m.budagavi}@samsung.com

***Abstract*—The Intra Block Copy (IntraBC) is a newly adopted tool in the HEVC extension for the screen content video coding. The IntraBC tool efficiently encodes repeating patterns in a picture. The current IntraBC scheme achieves about 1.0% bit-rate reduction on average and up to 4.3% bit-rate reduction on natural content video for a database consisting of 2K, 4K, and 8K sequences. In this paper, we propose to improve the IntraBC with a template matching block vector and a fractional search IntraBC. With these two tools, the gain on natural content video coding can be further improved by 0.5% on average and up to 2.0%.**

***Keywords*— HEVC, Intra Block Copy, fractional search, template matching.**

I. INTRODUCTION

The Intra Block Copy (IntraBC) mode is a newly adopted extension tool in High Efficient Video Coding (HEVC) for screen content coding (SCC) [1][2]. In the IntraBC scheme, a previous encoded block within the frame can be used as a predictor for the current block. This scheme works efficiently on screen content coding since usually there are many repeated patterns in the screen content video, and IntraBC can remove the redundancy much better than the traditional intra prediction (where the predictors only come from neighboring pixels). However, it has been shown that the gain of the current IntraBC scheme on natural content video coding (about 0.5% according to [3], and about 1.0% on a different test set of HD and UHD sequences that we use in our simulations) is much smaller than that on the screen content video. In this paper, we describe our proposed improvements on the IntraBC scheme to increase the coding performance on natural content videos.

In inter-frame motion estimation, the motion vector is encoded explicitly in the bit stream. To remove the redundancy between neighboring motion vectors (when it has uniform motion), a motion vector predictor is used to generate the motion vector differences. After prediction, the motion vector differences approximately follow the geometry distribution and the Exponential-Golomb coding [4] is used to encode the vector differences. A similar process is applied in the current IntraBC block vector (BV) coding. However, since the neighboring BVs in intra-frame coding may not have high correlations, the vector difference coding still needs a large amount of bits. To overcome this problem, [5] and [6] investigate the use of boundary template matching in H.264/AVC, so that the BV can be derived from previous reconstructed regions on the decoder side without encoding in the bit stream. It was pointed out that using some boundary matching schemes, the coding gain could be up to 20% in H.264/AVC intra-frame coding. In this paper, we investigate the use of boundary template matching in the IntraBC for HEVC.

In the current IntraBC, the prediction block is searched on a integer-pixel level. This is suitable for computer-generated graphics and can reduce the searching complexity and motion vector overhead bits. However, in camera-captured videos, it was pointed out that fractional search can achieve a good gain in inter-prediction [7]. Half-pel search was adopted in H.263 and MPEG-2, quarter-pel search was adopted in MPEG-4 and H.264/AVC, and a 1/12-pel search scheme was proposed in the early stage of HEVC standard [8]. In this paper, we investigate the use of fractional search in IntraBC for intra-prediction to increase its prediction accuracy. We also address related issues such as the padding in the interpolation and the entropy coding of the block vectors for the fractional search in the IntraBC.

The reminder of the paper is as follows: Section II discusses the template matching BV scheme. Section III discusses the fractional IntraBC search. Section IV presents experimental results of these two methods. Section V concludes this paper and describes the future work.

II. TEMPLATE MATCHING INTRA BLOCK COPY

As described in the introduction, the BVs in the neighboring blocks may not have high correlations, so that the vector differences after prediction are still large, resulting in large overhead bits for encoding the BV differences. Our experimental results in Table I show that omitting the BV difference bits, the overall BD-bitrate [9] reduction in Intra-frame coding can be up to -9.0%, which is very significant for the HEVC standard.

+ This work was supported by the Samsung Research America.

TABLE I. BD-BITRATE REDUCTION WHEN OMITTING OVERHEAD BITS FOR THE BV DIFFERENCES.

Sequences	BD-Bitrate reduction
BasketballDrill	-9.0%
BasketballPass	-3.1%
BlowingBubbles	-3.2%
BQMall	-4.1%
BQSquare	-8.8%
PartyScene	-7.1%
RaceHorses	-2.9%
RaceHorsesC	-4.9%

To reduce the overhead bits for the BV differences, there are three possibilities: 1) finding better BV predictors to reduce the BV differences, 2) finding more suitable entropy coding schemes for IntraBC, and 3) deriving the BV on the decoder side so that the transmitted overhead can be completely avoided. Obviously, these three methods can be used jointly. In this paper, we focus on 3), and use the boundary matching scheme to implicitly transmit the BV information in the IntraBC.

A. Boundary template matching

We use an inverse-L area adjacent to the coding block as the matching template, as shown in Fig. 1. In this way, all the matching pixels in both the encoder and the decoder sides are reconstructed prior to the current coding block. For the selection of the template width, we use a 2-pixel width, based on our testing results.

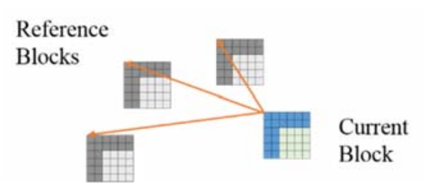

Figure 1. Current coding block (green) with its adjacent matching template (blue) and some reference blocks (gray). (Best view in color)

With the aforementioned template, we can estimate the similarity between the current coding block and the reference candidates based on the matching template difference (SAD is used in our experiments). It is likely that the reference block is similar to the current coding block when the boundary parts of these two blocks match well.

Obviously, this template matching method is not based on the cost of the actual coding block, and so, the template-matching-based BV may not be as good as the BV found in the traditional block-matching method. To guarantee the performance is not bad in the worst case, we use this template matching BV as an extra mode, instead of replacing all motion-estimated BVs in the IntraBC. Hence, one overhead flag is transmitted to the decoder to indicate if the template matching BV is used.

B. Matching template refinement

To improve the matching, we perform a refinement on a set of N template matching BV candidates with smallest SADs within the search range. Among these N BVs, the index i ($1 \leqslant i \leqslant N$) of the vectors with the minimum SADs for the *coding block* will be transmitted to the decoder side. In the decoder side, the identical set of BV candidates can be derived. After decoding the index i, the i-th candidate will be used as the final BV.

In [5], it is pointed out that using the average of some similar blocks can generate a better predictor. Hence, in this refinement scheme, we use the index N to indicate the averaging of all N candidates. In our experiments, we select N as 8.

C. Overhead encoding for boundary matching

In the proposed template matching scheme, two extra overhead flags are introduced, one is indicating whether the boundary matching method is used, and the other is the refinement index i. In our implementation, we use entropy coding to encode the indicating flag, and use 3-bit fixed-length coding for the refinement index. Our experimental results show that using the entropy coding for the indicating flag can reduce the overall encoding bitrate by 0.1%.

III. FRACTIONAL INTRA BLOCK COPY SEARCH

In the inter-frame fractional-pel search in the HEVC reference software HM-14.0 [10], the integer motion vector is derived based on the minimum sum of absolute difference (SAD) and then the half-pel refinement is applied around the integer motion vector position, followed by the quarter-pel refinement based on the half-pel refined position, as shown in Fig. 2. In the current HEVC standard, a DCT-based 7 or 8 tap filter is applied for the half and quarter pixel interpolation. The similar fractional search scheme of inter-frame motion estimation can be applied on the IntraBC. Some issues will be discussed as follows.

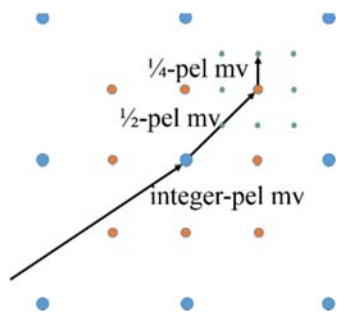

Figure 2. Three-step refinement for fractional motion vectors.

A. Padding in the interpolation

Since only the pixel within the current frame and reconstructed prior to the current coding block can be used as the prediction pixels, some pixels used in the 7 or 8 tap interpolation filter may not be available, as shown in Fig. 3. We investigated two methods to handle this situation: 1) skipping the fractional refinement when those corresponding interpolation pixels are not available, and 2) performing some padding schemes for the interpolation so that each integer position can perform a fractional refinement. Based on simulation results, skipping the refinement on some blocks reduces the overall gain of fractional search, therefore, we choose to implement the padding for the unavailable pixels.

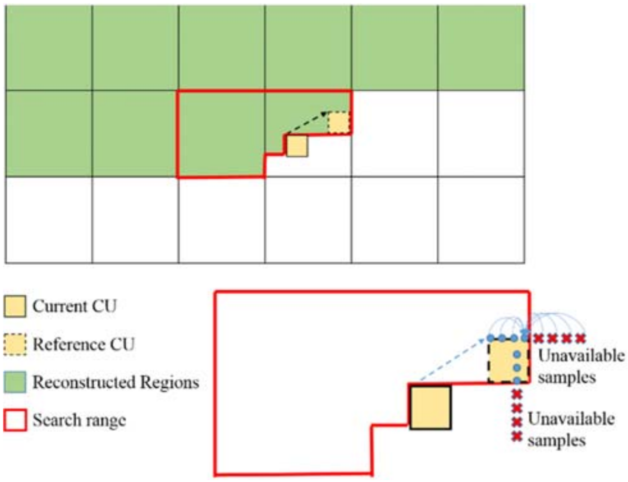

Figure 3. IntraBC search range and unavailable samples in half and quarter pixel refinement. (Best viewed in color)

Different padding schemes can be used such as repeating padding or symmetric padding. In the repeating padding, the unavailable pixel is padded with the same value of the nearest pixel which is available on the same row or column. In the symmetric padding, the unavailable pixel is padded with the pixel symmetric to padding boundary. Based on our simulation results, different padding schemes do not affect the coding gain much. Considering the simpler implementation of the repeating padding scheme, we use it in our experimental part. Note that this padding scheme is applied on both luma and chroma components.

B. Exp-Golomb coding of quarter-pel BV difference

Exp-Colomb coding with the order k ($\geqslant 0$) is a kind of universal code. The larger k, the closer the code-word is to the fixed-length coding. In the current IntraBC scheme, k is set to 1. However, with a quarter-pel accuracy BV, the BV and the BV difference will be 4 times larger than that with integer-pel accuracy, which increases the bits for BV by 8 bits (4 bits for each X and Y components). There are two methods to encode the fractional BV part (0, 1/4, 1/2 and 3/4) more efficiently. One is to increase k to 3 for the fractional part, which makes the Exp-Golomb coding suitable for fractional BVs. Another way is to encode the fractional BV part separately by a fixed-length coding, since those four parts are almost uniformly distributed. In our simulation, the later method is slightly better, and compared with the current Exp-Golomb coding in the IntraBC, it can reduce the average increment of bits for BV differences from 8 bits to 4 bits.

IV. Experimental Results

The proposed fractional search and template matching BV schemes are implemented in the HEVC reference software HM-15.0+RExt-8.0+SCM-2.0 (anchor), which includes the IntraBC in the SCC-profile. Some test conditions are in Table II. Note that in our simulations, to be consistent for all sequences, we have used 5 frames only as the computation time for 8K sequences is significant. From our simulations on several frames on lower resolution sequences, the behavior for intra tools can be very well approximated by the results from very few intra frames (one to five).

TABLE II. Test Conditions

Coding configuration	All-intra coding
QP	22, 27, 32 and 37 (main-tier)
IntraBC search range	Left and current LCU
Test sequences	4:2:0 YUV sequences with different resolutions (2K, 4K and 8K)
Number of frames	5

A. Results of the proposed tools

To compare the performance of our proposed tools, we perform the following tests:

(1) Compare the anchor without IntraBC to the anchor with IntraBC;

(2) Compare the anchor with IntraBC to template matching BV;

(3) Compare the anchor with IntraBC to fractional IntraBC;

(4) Compare the anchor with IntraBC to template matching BV + fraction IntraBC;

The BD-bitrate reduction percentages of the above tests are shown in Table III, where a negative number means coding gain. We highlight the numbers in Test 4, where our proposed coding tools provide more than 0.5% gain. Note that, as expected, our tools mainly provide gains when the IntraBC itself provides significant gains. On the other hand, for sequences such as Kimono, where IntraBC mode provides almost no gain, our tools also do not provide any further gain.

TABLE III. BD-bitrate Reduction on Different Tests

		Test 1	Test 2	Test 3	Test 4
2K (1920 ×1080)	Kimono	-0.0%	-0.1%	-0.1%	-0.1%
	ParkScene	-0.3%	-0.2%	0.0%	-0.2%
	Cactus	-1.5%	-0.8%	-0.5%	-1.3%
	BasketballDrive	-0.8%	-0.7%	0.1%	-1.1%
	BQTerrace	-2.2%	-1.5%	0.0%	-1.6%
4K (3840 ×2160, except Hotel 4096 ×2048)	chuno_s4	-0.1%	-0.2%	0.0%	-0.2%
	chuno_s31	-0.1%	-0.1%	0.1%	0.1%
	crownrun	-0.1%	0.0%	0.0%	-0.1%
	hotel	-0.9%	-0.8%	-0.5%	-1.4%
	Pku_girl	-0.2%	-0.3%	0.1%	-0.3%
	reed	-0.0%	0.0%	0.0%	0.0%
	4k_seq1	-0.0%	-0.1%	0.0%	-0.1%
	butterfly	-0.3%	-0.4%	0.1%	-0.4%
8K (7680 ×4320)	8K_seq7	-0.4%	-0.4%	0.2%	-0.4%
	8K_seq12	-1.1%	-0.5%	-0.1%	-0.7%
	8K_seq14	-4.3%	-0.4%	0.6%	0.1%
	8K_seq17	-0.1%	-0.1%	0.0%	-0.1%
	8K_seq18	-2.2%	-0.6%	0.2%	-0.6%
	DS_store	-3.4%	-0.4%	-1.6%	-2.0%
	2K average	-1.0%	-0.7%	-0.1%	-0.9%

	4K average	-0.2%	-0.2%	0.0%	-0.3%
	8K average	-1.9%	-0.4%	-0.1%	-0.6%
	All average	-1.0%	-0.4%	-0.1%	-0.5%

We have following remarks on these results:

(1) The BD-bitrate reduction of template matching BV is up to 1.5% and the gain of fractional search is up to 1.6%.

(2) Comparing tests (2), (3) and (4), it can be seen that for most sequences, the gains of fractional search and template matching BV are additive.

(3) The combined BD-bitrate reduction of our proposed improvements is 0.5% on average, and up to 2.0%. Considering the additional BD-bitrate reduction from IntraBC is about 1.0%, the overall BD-bitrate reduction of the improved IntraBC on natural content video coding is 1.5% on average.

(4) Considering there is only a small portion of blocks using the IntraBC mode in natural content video coding (most blocks are coded as intra prediction modes), our proposed tools has a significant gain on the IntraBC-coded blocks.

B. Hit ratio of our proposed tools

To demonstrate that our proposed tools on IntraBC is efficient, we plot the hit ratio for the sequence "Hotel" in Fig. 5. In this figure, the "regular IntraBC" indicates the blocks useing integer-pel IntraBC with explicitly transmitted BV. It can be seen that most IntraBC-coded blocks use our proposed tools. More than half of the IntraBC-coded blocks use the template matching BV scheme. This is consistent to the result that the template matching BV scheme has better performance than the fractional IntraBC.

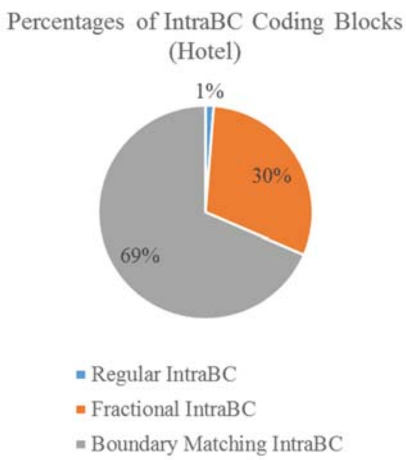

Figure 5. Percentage of IntraBC coding blocks using regular IntraBC (the current IntraBC scheme in the HEVC standard), fractional BV, and boundary matching IntraBC.

The usage of different IntraBC-coded blocks are shown in the frame (Fig. 6) as an example. It should be noted that the IntraBC blocks often appear at the object boundary. For the flat parts, the IntraBC scheme is not as good as the traditional intra prediction.

Figure 6. Different types of IntraBC-coded blocks in a frame in "BasketballDrill". Black is the current block being coded. (Blue: Fractional IntraBC, green: Boundary Matching IntraBC, red: Regular IntraBC. Best view in color).

V. Conclusion

In this paper, we present the fractional IntraBC and the template matching block vector for the current Intra Block Copy scheme in HEVC Screen Content Extension to make it more suitable for the natural content coding. The gain of these two tools is about 0.5% and can be up to 2.0%. The future work includes finding a better BV predictor and more suitable entropy coding for encoding the BVs in IntraBC.

References

[1] M. Budagavi and D.-K. Kwon, "AHG8: Video coding using Intra motion compensation," *Joint Collaborative Team on Video Coding (JCT-VC)*, JCTVC-M0350, 2013.

[2] R. Joshi, J. Xu, R. Cohen, S. Liu, Z. Ma and Y. Ye, "Screen Content Coding Test Model 2 Encoder Description (SCM 2)," *Joint Collaborative Team on Video Coding (JCT-VC)*, JCTVC-R1014, 2013.

[3] S. Lee, C. Park, E. Alshina, C. Kim, and K. McCann, "Ahg5: Recommended profiling of range extension coding tools," *Joint Collaborative Team on Video Coding (JCT-VC)*, JCTVC-Q0051, 2014.

[4] H. Malvar, "Adaptive run-length/golomb-rice encoding of quantized generalized gaussian sources with unknown statistics," *Data Compression Conference, 2006 Proceedings*. IEEE, 2006, pp. 23-32.

[5] T. Tan, C. S. Boon, and Y. Suzuki, "Intra prediction by averaged template matching predictors," *Consumer Communications and Networking Conference, 2007. CCNC 2007.* 4th IEEE. IEEE, 2007, pp. 405-409.

[6] C. Lan, J. Xu, F. Wu, and G. Shi, "Intra frame coding with template matching prediction and adaptive transform," *17th IEEE International Conference on Image Processing (ICIP), 2010*. IEEE, 2010, pp. 1221-1224.

[7] B. Girod, "Motion-compensating prediction with fractional-pel accuracy," *IEEE Transactions on Communications*, vol. 41, no. 4, pp. 604-612, 1993.

[8] K. McCann, W. Han, I. Kim, J. Min, E. Alshina, A. Alshin, T. Lee, J. Chen, V. Seregin, S. Lee, et al., "Samsung's response to the call for proposals on video compression technology," *Joint Collaborative Team on Video Coding (JCT-VC)*, JCTVC- 124, 2010.

[9] G. Bjontegaard, "Calculation of average psnr differences between rd curves," *VCEG-M33 ITU-T Q6/16*, Austin, TX, USA, 2-4 April 2001, 2001

[10] JCTVC, "Hevc test model (hm)," http://hevc.kw.bbc.co.uk/git/w/jctvc-hm.git/commit/1204fc83b1f917237624e9ec8c3c4d7bebab58d5, 2013, [Online; accessed 4-Nov-2013].