Intra Line Copy for HEVC Screen Content Coding

Tsui-Shan Chang, Chun-Chi Chen[1], Ru-Ling Liao, Che-Wei Kuo, Wen-Hsiao Peng[2]
Department of Computer Science, National Chiao Tung University
1001 Ta-Hsueh Rd., Hsinchu 30010, Taiwan
Email: [1]cheerchen.cs98g@g2.nctu.edu.tw, [2]wpeng@cs.nctu.edu.tw

***Abstract*—This paper presents an intra line copy mode, which introduces a finer-granularity block partitioning structure for the intra copying technique, to further improve the coding efficiency for screen content. This mode first divides a prediction block horizontally or vertically into 1-pixel-wide lines and then operates intra copying line-by-line from the reconstructed pixels in the current frame. It shares some parallels with the notion of string matching widely used for lossless data compression. It, however, is more compliant to the current HEVC design and is parallel-friendly. This paper shall detail its various design issues, including the design of a fast algorithm based on hashing techniques and the impact of search window size on its performance and complexity trade-off. Experimental results show that the proposed mode can offer 2.7-12.9% of BD-rate reductions, with 26% increase of encoding runtime while keeping the decoding runtime unchanged, when integrated into the SCM-1.0 test model.**

I. INTRODUCTION

In January 2014, after the finalization of HEVC Range Extensions (RExt) standard [1], the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group jointly issued a Call-for-Proposals on coding of screen content which targets at enabling HEVC the compression capability for non-camera content applications. Responses to the call have shown significant reduction of bit-rate over 50% while still offering an equal level of video quality as comparing with the draft HEVC RExt standard.

Typical types of screen content generally cover text, computer-generated graphics, natural images, or an arbitrary mixture of them, and thus the characteristics of such content are very different from the camera-captured content. For instance, one-pixel-wide lines, sharp edges and repeated patterns are frequently occurred in most screen contents. Conventional video coding techniques such as intra prediction and DCT-based transform coding cannot effectively produce a sparse representation for signals with such high dynamics. A report [2] has shown evidences that the lossy coding performance of the HEVC reference software is inferior to some techniques widely used for lossless data compression.

The recent approaches proposed in literature for handling such content can be categorized into three typical sets, including transform skipping, palette coding and string matching techniques. Transform skipping [3] is the simplest solution among them, which bypasses the transform coding and treats the residual samples as they were the transform coefficients for quantization. However, the essence that leads inefficient coding performance is the inadequate use of intra prediction which is designed mainly for camera-captured content but not for screen content. Accordingly, two alternative approaches turn to make the best of the signal characteristics of typical screen contents, that is, the frequent occurrence of periodical patterns containing a few colors.

The palette coding [4] bypasses the intra prediction in addition to skipping the transform, and redesigns the quantization and entropy coder. The mapping from quantization indices to sample values is explicitly signaled, and a method similar to the sliding window Lempel-Ziv coding is used as instead for coding of quantization indices. On the contrary, the string matching approaches [5]-[6] tend to improve the performance of intra predictions by introducing finer-granularity block partitions and using the sliding window Lempel-Ziv coding for each partition to search repeated patterns from valid search regions inside the current frame. The intra block copy (IntraBC or IBC) [7]-[8] is one of its variants, which further aligns the block partitions with that of inter prediction modes.

In this paper, we propose an intra line copy mode which provides a new block partition structure at finer-granularity levels than those being used in HEVC to improve the prediction performance of intra copying. When using this mode, the current prediction unit is split equally into a multiple of 1-pixel-wide rows or columns (which are referred to as lines), and then the intra copying is performed in a line-by-line fashion. Similar to IntraBC, this mode also refers to the most-recently decoded block vector (BV) for block vector prediction. Besides, it is observed that BVs are found frequently to be vertical or horizontal displacement (e.g. x or y component equal to zero), and the residual signals of this mode tend to be dropped. Accordingly, a fast BV search algorithm based on 1-D displacement search and the cyclic redundancy check (CRC) hash function is thus proposed. Furthermore, the memory access bandwidth of this mode for fetching reference samples from off-chip memory is also discussed, which leads to a constraint on the size of the search range for the intra line copy mode. The experimental results show that the intra line copy mode can achieve 2.7-12.9% of BD-rate savings, with only 26% encoding time increase and negligible impact on decoding runtime, when integrated into the SCM-1.0 test model [9].

The rest of this paper is organized as follows. Section II provides an overview of the string matching based techniques currently under evaluation in the latest screen content coding standard. In Section III, we propose the intra line copy mode and a fast algorithm based on hash technique for it. Section IV provides the experimental results and analyses of memory access bandwidth of the proposed mode. Section V concludes this paper.

II. OVERVIEW OF STRING MATCHING AND ITS VARIANTS

This section will introduce the basics of the string matching and several of its variants, including pseudo-2D-matching coding [5]-[6], reconstruction-based dictionary coding [6], and intra block copy [7].

A. Basics of String Matching

The string matching, widely known as the sliding window Lempel-Ziv coding, is a universal variable-to-variable-length coding scheme, which achieves compression by:

(1) decomposing a string into a disjoint set of substrings;

(2) representing each substring by indicating its length and the associated displacement to where a repeated pattern occurs in the previously decoded substrings (referred hereafter to as the search buffer).

The length of each substring is determined by the longest perfect match that can be found in the given search buffer. Fig. 1 demonstrates an example that a substring, "abra", is perfectly represented by the matching length and the displacement equal to 4 and 6, respectively. Besides, a new substring is explicitly coded if no repeated pattern can be found in the past. Once a substring is coded, it will then serve as a part of the search buffer for its immediately subsequent substring. The aforementioned process will not stop until all the substrings have been coded. Obviously, the string matching is expected to perform extremely efficient if periodic patterns occur frequently within a string.

B. Pseudo-2D-Matching Coding

As periodical patterns can occur frequently in some typical types of screen content (e.g. text and graphics as in Fig. 2), the Pseudo-2D-Matching (P2M) [5]-[6] coding mode applies the notion of the 1-D string matching to efficiently represent such 2-D contents. As depicted in Fig. 3, the 2-D samples in the current frame are firstly scanned block after block according to the coding order and the samples within each block are then rearranged separately into vertically- and horizontally-scanned 1-D strings. Each block can choose adaptively between the two scanning patterns. Lastly the 1-D string matching introduced in Section II.A can thus be applied.

The major problems of this approach are threefold. First, the design is less parallel-friendly because the starting position of a substring is not known until the length and displacement of its previous substring are determined. A sequential encoding of substrings is thus unavoidable.

Second, the 2-D block structure cannot be reserved due to the 1-D scanning. For example, Fig. 4 shows that the longest perfect match of "abra" cannot be found from the given 1-D search buffer. Third, additional decoding process and CABAC logics are required to support this new mode.

C. Reconstruction-based Dictionary Coding

To preserve the 2-D block structure when using string matching for coding of screen content, the reconstruction-based dictionary [6] coding mode was proposed. As shown in

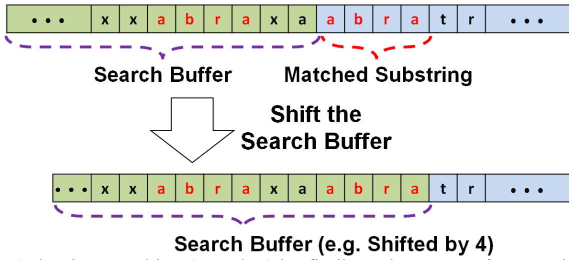
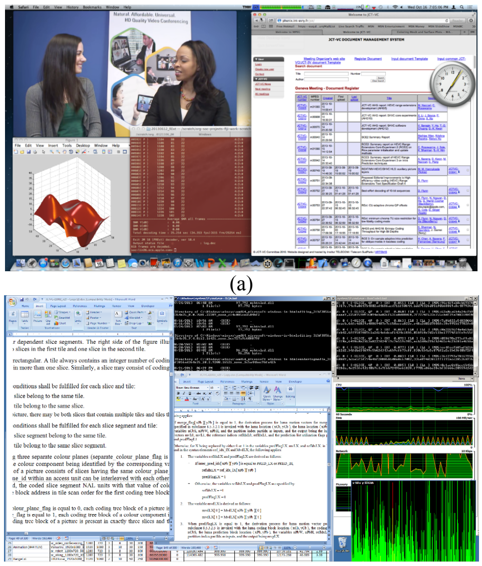


Fig. 1. Substring matching (e.g. abra) by finding a longest perfect match in the given search buffer. Then, the coded substring is concatenated to the end of the given search buffer. The resulting buffer will be shifted if it exceeds the maximum size of the search buffer.



(a)

(b)

Fig. 2. Typical types of screen content. (a) MissionControlClip3; (b) sc_desktop.

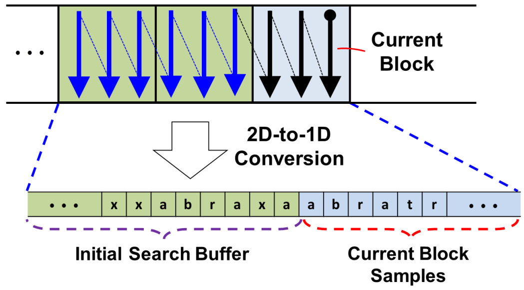


Fig. 3. 2-D blocks to 1-D string conversion. The green and the blue areas represent respectively the reconstructed samples and the samples to be coded.

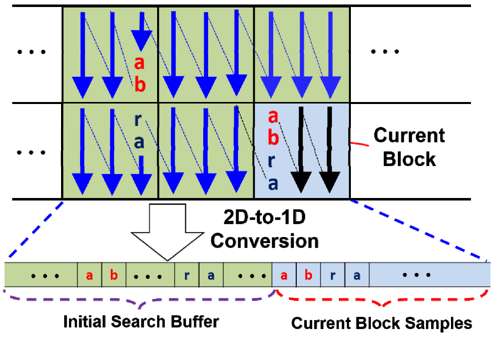


Fig. 4. The P2M coder does not preserve 2-D block structure. The example shows that the current substring, "abra", cannot be found in the search buffer even though such pattern once occurred in the past.

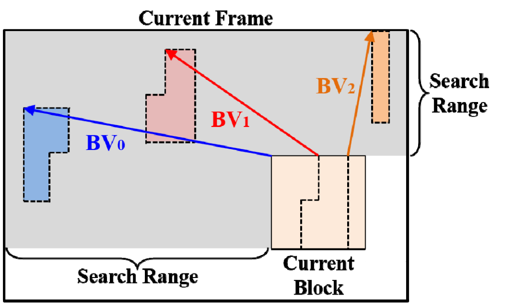


Fig. 5. Reconstruction-based dictionary coding finds the perfect match of a substring from the reconstructed region within the current frame rather than searching from the 1-D search buffers.

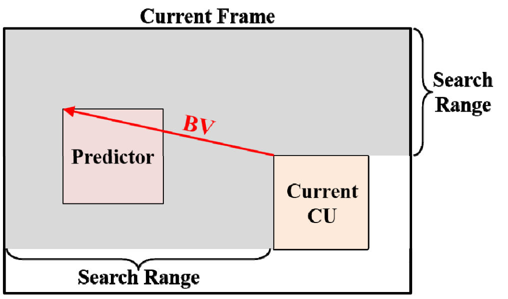


Fig. 6. IntraBC performs intra copying which fetches reference pixels from the reconstructed area of current frame.

Fig. 5, this method directly finds the perfect match of a substring from the reconstructed region within the current frame rather than searching from the 1-D search buffers. Each substring is represented by using its length and a 2-D displacement indicating where a perfect match exists in the current frame. Otherwise, a substring is explicitly signaled if the perfect match does not exist. It is noted that pixels within the current CU cannot be used as reference.

Conceptually, this method can be viewed as being operationally equivalent to:

- offering the block-based video codecs a more flexible block-partitioning manner than the commonly-seen square, symmetric, and asymmetric block partitions;
- performing motion-compensated prediction for these block partitions except using the current frame as reference.

Although the 2-D block structure can be preserved, the same issues as P2M's regarding parallel-friendly design and extra decoding logics are not resolved yet

D. *Intra Block Copy*

To avoid the design issues described in Section II.C, the intra block copy (IntraBC or IBC) technique [7] introduces two modifications to the reconstruction-based dictionary coder while still keeping comparable coding performance to it. The modifications are described as follows:

- restricting the length of each substring to being exactly the same as the number of pixels of its associated block (e.g. typically a coding unit (CU));
- replacing the matching criterion by using a rate-distortion metric (e.g. the sum of absolution differences plus the overhead for signaling the 2-D displacement), and reusing the existing logics of a video codec for residual coding and 2-D displacement coding.

As can be seen from Fig. 6, the predictor of a CU is fetched from the given search area inside the current frame and the associated 2-D displacement (referred hereafter to as block vector or BV) is sent in the same way as that of motion vector coding. Since there is only one substring per block, the sequential encoding of substrings no longer exists. No extra burden for the decoder will be introduced to support this new coding tool, because the string matching can be implemented by reusing the existing hardware logics for inter modes (e.g. motion vector coding and motion compensation) and residual coding. Only an additional flag is required for signaling the use of this new mode.

The IntraBC mode is currently adopted in the test model (SCM-1.0) [9] for the screen content coding extensions to the HEVC video coding standard.

E. *Summary*

To summarize, IntraBC is more compliant and parallel-friendly to the current HEVC design than the other string-matching-based approaches due in that no extra decoding burden will be introduced for supporting this mode. In our previous work [8], it reveals that IntraBC tends to be enabled more frequently at smaller block levels and extends the IntraBC to prediction uint (PU) levels including 2NxN, Nx2N, and NxN partitioning types. This extension is also adopted into SCM-1.0. In the next two sections, we will explore more regarding how a finer-granularity block partition structure, termed as line, impacts the video codec in terms of complexity and coding performance.

III. Intra Line Copy

In this section, we present the framework of the proposed intra line copy and introduce a hash-based BV search to speed up the BV search based on the observations on the BV differences and prediction residuals.

A. *Framework of Intra Line Copy*

The intra line copy (IntraLC or ILC) mode extends the basic unit for IntraBC from blocks to lines, which divides a PU row-wisely or column-wisely into multiple 1-pixel-wide lines. Then it performs the intra copying operation line-by-line (Fig. 7) from the same full-frame search range as specified for IntraBC. Like an IntraBC block, each line has its own BV shared across three color components. A mode flag is present at CU level for the encoder to choose the IntraBC mode or the IntraLC mode adaptively, and the splitting manner for lines (e.g. row or column) is indicated at CU level. The resulting patterns for lines are summarized in Fig. 8. Once the prediction of each line is done, the residual signals of the associated CU are transformed, quantized, and entropy coded in the same way as IntraBC does.

For the BV coding, obviously a finer-granularity block partitioning can perform much better than a coarser one in terms of prediction performance. The price paid, however, is the increased number of block vectors to be sent for a CU. In

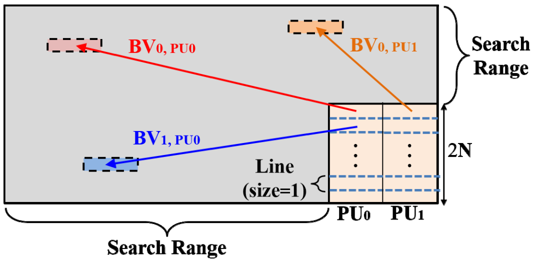


Fig. 7. The intra copying operation is similar to that of IntraBC except that the basic unit applied to this operation is a line instead of a block.

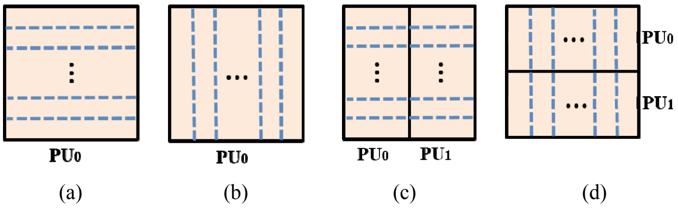


Fig. 8. The supported PU types and the associated line patterns, including (a)(b) 2Nx2N PU with horizontal and vertically splitting, respectively; (c) Nx2N PU with horizontal splitting; (d) 2NxN PU with vertical splitting.

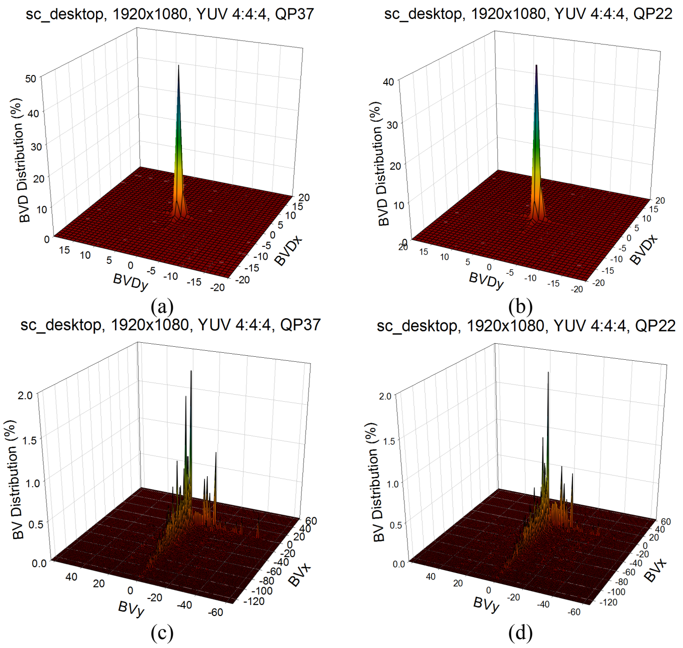


Fig. 9. (a)(b) BVD distribution and (c)(d) BV distribution generated under the All-Intra test conditions at QP37 and QP22 for sc_desktop. In (a) and (b), roughly 93% and 86% of the BVDs can be bounded within a 16x16-pixel area pointed to by the associated BV predictor of each line.

the worst-case scenario, there are 16 BVs to be coded at the 8x8 CU level, which is 4 times as many as those used for the NxN IntraBC. In this paper, the same BV coding scheme as for IntraBC is re-used, that is to predict the BV by using the most-recently decoded BV and code the BV difference (BVD) in the same way as for MVD coding. Later, in Section III.B.1, we will justify the effectiveness of this BV coding scheme.

The IntraLC, as a result, inherits some advantages from IntraBC and string matching that:

- it offers a subset of the partitioning structure supported by the reconstruction-based dictionary coding while still capable of searching each substring in parallel;
- it remains nothing changed for the BV coding and residual coding specified for IntraBC.

Thus, nearly no extra burden on decoder is introduced to support this new coding tool.

B. Hash-based BV Search for Lines

As the encoding time for the IntraLC mode can be significantly increased due to full-frame search, in this section we will analyze (1) the distribution of BVD and BV and (2) the intensity value of residual signal. The observations on them will then serve as the basic principle to elaborate a fast algorithm for BV search. The simulations are carried out using the SCM-1.0 software [9] at two different rate points (QP=22 and 37) and the All-Intra configuration. The results are based on 1-second encoding with exhaustive BV search over the full-frame search window for each line.

B.1. Distribution of BVD and BV

Fig. 9 (a) and (b) plot the BVD distribution of the sc_desktop sequence. These BVDs are collected only for the blocks coded by using the IntraLC mode. The Z-axis of these plots represents the count of a BVD. As can be seen, it is quite obvious that most of the BVDs are located in a 16x16-pixel area and are centered at the origin (0, 0). Besides, these results also justify the effectiveness of applying the existing BV prediction scheme to the proposed IntraLC mode (Section III.A).

For the distribution of BVs shown in Fig. 9 (c) and (d), we observed that most of the BVs are horizontal or vertical displacements (denoted as 1-D BVs), the lengths of which are generally within a 120-pixel-wide range to the left and a 60-pixel-height range on the top. Obviously, the coverage is roughly the same as the range covered by 2 CTUs[1]. This observation is still useful because the BVDs of some 1-D BVs are out of the aforementioned 16x16-pixel region.

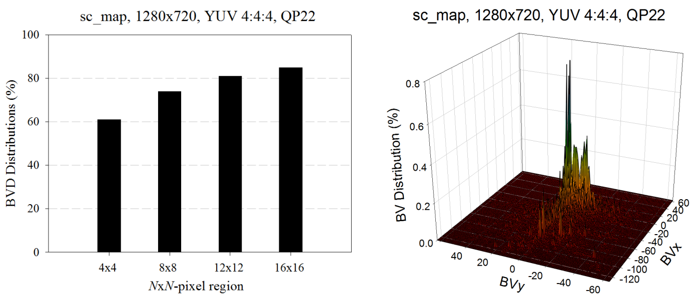
As a result, we will adopt two BV searching schemes to the design of the fast algorithm, one of which is to search over the entire 16x16-pixel area pointed to by the BV predictor, and the other of which is to search vertically and horizontally within the 2-CTU search range.

B.2. Residual Signal

However, there are still some BVDs and BVs outside the aforementioned search space. For example, the sc_map sequence, roughly 15% of its BVDs are observed out of the pre-defined 16x16-pixel area (Fig. 10). We approach this problem by resorting to investigating the residual signal for the blocks coded by using the IntraLC mode. As can be seen from Fig. 11, most of the residual signals need not be sent. Rather, most of the lines can find their own perfect matches or good matches based on their rate-distortion performance.

Based on the above observations, we intend to speed up the BV searching process by restricting the search space to a small set of lines which includes all the perfect matches of the current line. Accordingly, we adopt a hash technique to group

[1] To be precise, the *N* CTUs (or *N*-CTU) denotes one current CTU plus (*N*-1)*64-pixel-wide region to the left, where *N* is a positive integer.



(a) (b)

Fig. 10. (a) BVD and (b) BV distributions generated under the All-Intra test conditions at QP22 for sc_map.

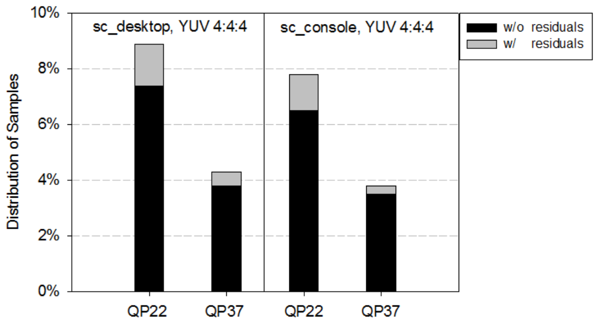


Fig. 11. Mode distribution of the intra line copy with residuals and the one without residuals for sc_desktop and sc_map.

all the identical lines. To make the number of lines in each group not too large, similar lines are needed to be classified into different groups due to the correlation of signals in a frame. As a result, the cyclic redundancy check (CRC) hash function is incorporated [6] into the fast BV searching algorithm. Before each frame starts encoding, a preprocessing stage is required to compute the hash values, one frame at a time, for each line by taking their original samples as the input to the hash function. Specifically, all the samples of each line are concatenated to form a binary string for computing CRC value. Four hash tables are in use for 1x4, 4x1 1x8 and 8x1 line partitions. The lines which share the same hash values are all collected by using a linked list. It is noted that the hash function offers a many-to-1 mapping, so if a perfect match of the current line exists, it will never be missed in the search process.

Therefore, the BV search can be as easy as picking up the one, which can produce the best prediction or rate-distortion performance, from the candidates that share the same hash value with the to-be-searched line. The search space is greatly reduced from full frame to only several reference lines.

B.3. Summary

To summary, the hash-based BV search algorithm for lines is described step by step as the follows:

1. searching within a 16x16-pixel area which is centered at the pixel position pointed to by the BV predictor;
2. searching horizontally and vertically inside the available part of the 2-CTU search range;
3. searching from a candidate list formed by all the reference lines which share the same hash value with the current line.

TABLE I. Y/G BD-RATE REDUCTION AND RUNTIMES FOR INTRALC WITH FULL-FRAME SEARCH BUFFER.

Configuration	AI	RA	LB
RGB, text & graphics with motion, 1080p	-12.9	-7.0	-3.8
RGB, text & graphics with motion,720p	-6.0	-4.6	-2.5
RGB, mixed content, 1440p	-3.3	-2.3	-1.5
RGB, mixed content, 1080p	-6.0	-4.2	-2.0
YUV, text & graphics with motion, 1080p	-12.0	-6.3	-3.2
YUV, text & graphics with motion,720p	-4.7	-3.8	-1.6
YUV, mixed content, 1440p	-2.7	-1.7	-0.6
YUV, mixed content, 1080p	-5.2	-4.0	-1.4
Enc Time[%]	126	104	103
Dec Time[%]	99	102	102

TABLE II. COMPARISON OF INTRALC, P2M AND RECONSTRUCTION-BASED DICTIONARY CODING WITH FULL-FRAME AND ALL-INTRA CONFIGURATION.

Algorithm	IntraLC	P2M	Recon.
RGB, text & graphics with motion, 1080p	-12.9	-7.7	-13.3
RGB, text & graphics with motion,720p	-6.0	-3.9	-6.7
RGB, mixed content, 1440p	-3.3	-0.3	-1.7
RGB, mixed content, 1080p	-6.0	-0.7	-4.6
YUV, text & graphics with motion, 1080p	-12.0	-8.2	-11.4
YUV, text & graphics with motion,720p	-4.7	-3.4	-5.4
YUV, mixed content, 1440p	-2.7	-0.3	-1.5
YUV, mixed content, 1080p	-5.2	-0.7	-4.0
Enc Time[%]	126	112	132
Dec Time[%]	99	136	96

Besides, according to [8], we additionally let the encoder skip evaluating this mode at the smallest CU level.

IV. EXPERIMENTAL RESULTS

In this section, the experimental results of the IntraLC mode are provided, followed by a comparison with this mode, P2M and reconstruction-based dictionary coding. The measurement of the worst-case memory access bandwidth of IntraLC is also provided, the results of which lead to a configuration of limiting the search range of IntraLC to 2 CTUs. We also enhanced the BV search for 4x4/8x4/4x8 IntraBC blocks with the intention to justify the necessity of line-based partitioning.

Experiments were conducted using the common test conditions [11] at 4 QP values (22, 27, 32, 37) and the HM-14.0+RExt-7.0+SCM-1.0 software [9].

A. Coding Results of Intra Line Copy

TABLE I shows the coding performance of intra line copy with full-frame search range. In the All-Intra (AI) cases, up to 12.9% BD-rate saving is observed with 26% encoding time increase. When the inter prediction is involved, roughly 2.0-7.0% and 1.0-4.0% coding gains are obtained with negligible encoding and decoding runtime increase for Random Access (RA) and Low Delay B (LB) configurations, respectively.

B. Comparison with P2M Coder and Reconstruction-based Dictionary Coder

TABLE II summarizes the coding results of IntraLC, P2M and reconstruction-based dictionary coding. From the table, two observations can be made. First, the P2M coder performs less efficiently than the others due to the 2-D to 1-D scanning that loses the structure of 2-D block patterns. Second, the IntraLC mode and the reconstruction-based dictionary coding are basically comparable to each other, except that the former is especially efficient for coding of mixed-content sequences. Moreover, the IntraLC mode is more compliant with the current design of HEVC and IntraBC than that of the two string matching based approaches. Nearly no change to the existing decoder design is required to support this coding tool.

C. Mode Distribution

Fig. 12 charts the mode distributions of the IntraLC by the average spatial coverage (in units of pixels) coded by using this prediction mode. From the figures, two observations can be made. First, it is quite intuitive that small block partitioning types, such as lines, tend to be enabled more in high-rate cases. Second, for the sequences (e.g. sc_desktop) of class TGM (text & graphics with motion), although the average spatial coverage (generally less than 10%) of the IntraLC mode is much lower than that of the Intra (70%+) and IntraBC modes (roughly 20%), significant BD-rate reduction (12.9%) can still be achieved.

As the IntraLC mode shares parallels with the notion of the reconstruction-based dictionary coding, we additional analyze how they behave differently based on their mode distributions. When comparing figure (a) with (c), it is observed that the reconstruction-based dictionary coder occupies almost 20-30% pixels area of the sequences (e.g. sc_desktop) of class TGM. However, as summarized in TABLE II**,** its coding performance is not differentiated too much as compared with the IntraLC mode. Even in some cases (e.g. the mixed content sequences -- MissionControlClip3 for example), the IntraLC mode outperforms the reconstruction-based dictionary coding mode in terms of coding gains while fewer pixels are coded by using the former one.

Fig. 13 visualizes the blocks coded by using IntraLC mode. It is observed that this mode tends to be enabled more frequently on blocks with, especially, text and thin lines. This is because such patterns are not easy to be accurately predicted by using large prediction units such as the IntraBC. The IntraLC mode thus takes the place, resulting in achieving noticeable improvement on the coding performance for screen content .

D. Coding Performance Relative to Improved SCM-1.0

To see whether it is necessary to introduce new partitioning types (e.g. 1x4, 4x1, 8x1 lines) to the SCM software, this section will provide the coding performance on top of an the SCM-1.0 software with some encoder-only change.

As can be seen from TABLE III, the SCM-1.0 software applies distinct encoder-only constraints on the search range to different PU types for the IntraBC mode, e.g. 2-CTU search range for 8x4, 4x8, and 4x4 PUs and full-frame search range for 8x8 and 16x16 PUs. As mentioned in [8], the IntraBC mode tends to be enabled more at smaller PU level, so 8x4, 4x8, and 4x4 PUs still have potential to be improved. So we apply a straightforward change to IntraBC that the search range of 8x4, 4x8, and 4x4 PUs is extended from 2 CTUs to full frame.

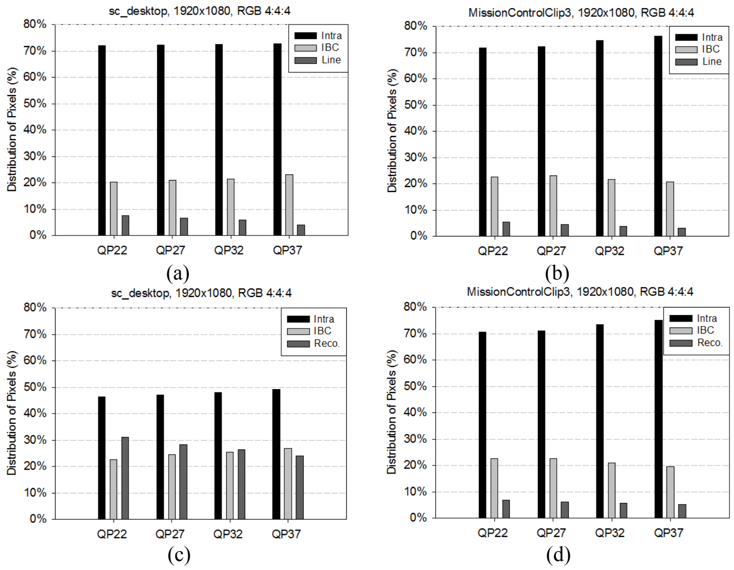


Fig. 12. (a)(b) Mode distribution of intra prediction, IntraBC and intra line copy in AI case. (c)(d) Mode distribution of intra prediction, IntraBC and reconstruction-based dictionary coding in AI case.

TABLE III. The Difference Of Search Range Between The SCM-1.0 And The Improved One.

SCM-1.0			
CU Size	2Nx2N	2NxN/Nx2N	NxN
8x8	2-D, Full-frame	2-D, 2-CTU	2-D, 2-CTU
16x16	1-D, Full-frame	(n/a)	
Others	(n/a)	(n/a)	
Improved SCM-1.0			
CU Size	2Nx2N	2NxN/Nx2N	NxN
8x8	2-D, Full-frame	2-D, Full-frame	2-D, Full-frame
16x16	1-D, Full-frame	(n/a)	
Others	(n/a)	(n/a)	

TABLE IV. Y/G BD-rate Reduction and Runtimes for IntraLC Relative to the Improved SCM-1.0.

Configuration	AI	RA	LB
RGB, text & graphics with motion, 1080p	-10.7	-5.9	-3.2
RGB, text & graphics with motion,720p	-5.1	-3.9	-2.2
RGB, mixed content, 1440p	-2.9	-2.3	-1.3
RGB, mixed content, 1080p	-4.1	-3.1	-1.4
YUV, text & graphics with motion, 1080p	-9.6	-5.2	-2.6
YUV, text & graphics with motion,720p	-3.6	-3.1	-1.1
YUV, mixed content, 1440p	-2.2	-1.5	-0.9
YUV, mixed content, 1080p	-3.3	-2.8	-0.7
Enc Time[%]	126	104	103
Dec Time[%]	98	99	99

Results summarized in TABLE IV are produced on top of the improved SCM-1.0 software. It is noted that the above improvement is only applied to the IntraBC mode, and thus the design of the IntraLC mode is still unchanged. The observations are generally similar to those delivered in Section IV.A. When compared with the results provided in TABLE I, the drop (1-2%) of coding gain is not obvious. The results of the AI, RA, and LB cases not only show moderate coding gains but also justify the necessity of the line-based partitioning in spite of the involvement of the small partitioning types (4x4/8x4/4x8 PUs) for intra copying.

TABLE V. WORST-CASE MEMORY ACCESS BANDWIDTH FOR BI-PREDICTION, INTRABC AND INTRA LINE COPY

	MxN	*L*	*mxn*	*P*
Bi-prediction	8x8	8	4x2	10
			8x2	12
			4x4	12.5
Intra Block Copy	4x4	1	4x2	3
			8x2	6
			4x4	4
Intra Line Copy	1x4	1	4x2	6
			8x2	12
			4x4	8

TABLE VI. COMPARISON OF INTRALC, P2M AND RECONSTRUCTION-BASED DICTIONARY CODING WITH 2-CTU SEARCH BUFFER AND AI CONFIGURATION.

Algorithm	IntraLC	P2M	Recon.
RGB, text & graphics with motion, 1080p	-9.4	-6.1	-5.2
RGB, text & graphics with motion,720p	-4.4	-3.1	-1.8
RGB, mixed content, 1440p	-2.7	-0.3	-0.2
RGB, mixed content, 1080p	-3.6	-0.7	-0.7
YUV, text & graphics with motion, 1080p	-8.8	-6.9	-4.0
YUV, text & graphics with motion,720p	-3.3	-3.1	-1.0
YUV, mixed content, 1440p	-2.1	-0.3	-0.2
YUV, mixed content, 1080p	-3.0	-0.7	-0.5
Enc Time[%]	127	105	110
Dec Time[%]	99	137	98

E. Worst-case Memory Access Bandwidth

Although the IntraLC offers a higher variety of block partitioning structure than that of the IntraBC, the impact on the memory access bandwidth for supporting this variety should also be considered to justify the feasibility of implementing a coding tool into real codec designs.

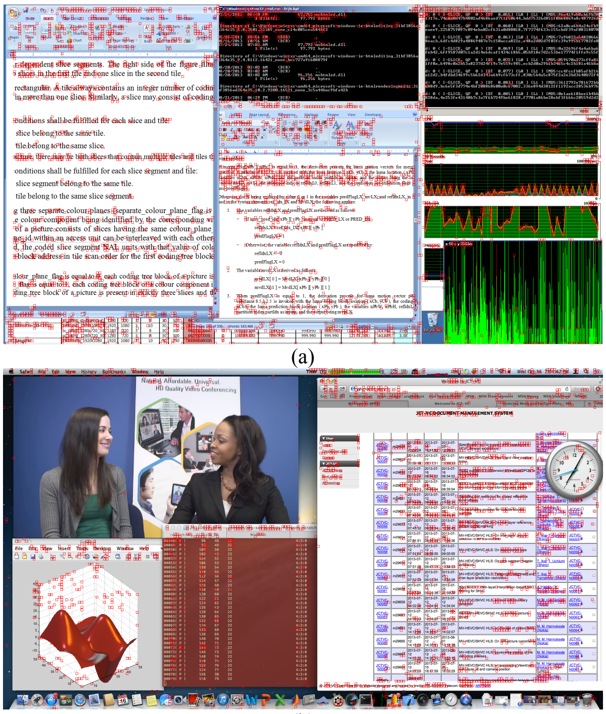
E.1. Memory Access Bandwidth

The per-pixel memory access bandwidth (denoted by *P*) [10] with respect to various memory access patterns (i.e. 4x2, 8x2, and 4x4) in the worst case are measured by using Eq. (1):

$$P = \frac{\left\lceil \frac{m-1+M+L-1}{m} \right\rceil \cdot \left\lceil \frac{n-1+N+L-1}{n} \right\rceil \cdot m \cdot n}{M \cdot N}\ , \qquad (1)$$

where *M*x*N* denotes the size of a target block, *m*x*n* represents the size of a given memory access pattern, and *L* is the interpolation filter length (e.g. *L*=8 and 1 indicating 8-tap interpolation filtering for inter modes and no interpolation filtering, respectively). Fig. 14 depicts an example for the case of inter prediction, and it shows that more pixels in addition to the effective pixels are needed to be fetched since the sub-pel interpolation is used and the fetched data is needed to be aligned to the basic unit of memory compression (memory pattern). We assume that all the reconstructed samples are stored in an off-chip memory space, so the intra copying operation will inevitably occupy the memory access bandwidth when fetching samples as reference. As the 4x4 IntraBC and 8x8 bi-prediction require the highest memory access bandwidth respectively for All-Intra and Inter cases, TABLE V summarizes the *P* values of them in comparison with that of the 1x4 line.

For Inter cases (RA and LC), the IntraLC does not impact the worst-case bandwidth. However, for the All-intra case



(a) (b)

Fig. 13. Visualization of the intra line copy mode distribution (red blocks) for (a) sc_desktop (RGB 4:4:4) and (b) MissionControlClip3 (RGB 4:4:4).

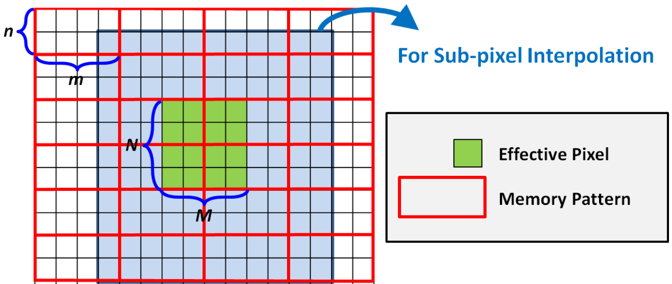


Fig. 14. An example of fetching pixels for inter prediction. The green area represents the *M*x*N* target block which contains the effective pixels, and the blocks with red-line boundaries are the *m*x*n* memory access pattern.

(AI), it doubles the worst-case bandwidth which is by no means acceptable in practice. We alleviate this impact by restricting the search range to be within *N* CTUs (where *N* is set equal to 2 in this paper) and allocating a dedicated on-chip buffer to store the most-recently reconstructed samples for intra copying. The buffer size is as large as the size of *N* CTUs (e.g. $2\text{x}64^2\text{x}3$=24 KBytes for *N*=2). Therefore, the modification can effectively prevent the IntraLC from worsening the worst-case memory bandwidth in the intra case.

E.2. Coding Results of the Constrained Cases

As the P2M and reconstruction-based coding have the same issue on memory access bandwidth, the aforementioned solution in the previous section is also applied. TABLE VI summarizes the results. The same observations described in Section IV.B can be drawn even though the search range is constrained to 2 CTUs. When comparing the TABLE II and TABLE VI, roughly 1-3% loss of BD-rate savings among the three schemes can be observed. Although the loss is inevitable, the IntraLC can still offer moderate BD-rate reductions of 2.1-9.4% relative to the anchors.

V. CONCLUSIONS

In this paper, an IntraLC mode is proposed. Various aspects of its design regarding the hash-based BV search for lines and memory access bandwidth are discussed. The results show that most of the BVs for lines have one component equal to zero and the prediction residuals tend to be dropped. The 2-CTU IntraLC is slightly inferior to the full-frame IntraLC in terms of BD-rate numbers, but it can be completely free from accessing off-chip memories for fetching references. As compared with the string matching related methods, this mode performs better especially for mixed contents and is much more compliant with the current design of the HEVC standard.

REFERENCES

[1] D. Flynn, M. Naccari, C. Rosewarne, K. Sharman, J. Sole, G. J. Sullivan, and T.Suzuki, "High Efficiency Video Coding (HEVC) Range Extensions Text Specification: Draft 7," ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCTVC-Q1005, March 2014.

[2] H. Yu, X. Wang, and J. Ye, "AHG8: More Investigation on Screen Content Coding," ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCTVC-M0320, April 2013.

[3] J. Sole, R. Joshi, and M. Karczewicz, "RCE2 Test B.1: Residue Rotation and Significance Map Context," ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCTVC-N0044, July 2013.

[4] P. Onno, X. Xiu, Y.-W, Huang, and R. Joshi, "Suggested Combined Software and Text for Run-based Palette Mode," ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCTVC-R0348, June 2014.

[5] T. Lin, X. Chen, and S. Wang, "Pseudo-2D-matching based Dual-coder Architecture for Screen Contents Coding," IEEE Int'l Conf. on Multimedia and Expo Workshops, July 2013.

[6] B. Li and J. Xu, "SCCE4: Results of Test 3.1," ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCTVC-R0098, July 2014.

[7] C. Pang, J. Sole, L. Guo, M. Karczewicz, and R. Joshi, "Non-RCE3: Intra Motion Compensation with 2-D MVs," ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCTVC-N0256, July 2013.

[8] C.-C. Chen, X. Xu, R.-L. Liao, W.-H. Peng, S. Liu, and S. Lei,, "Screen Content Coding using Non-square Intra Block Copy for HEVC," IEEE Int' l Conf. on Multimedia and Expo, 2014.

[9] R. Joshi, J. Xu, R. Cohen, S. Liu, Z. Ma, and Y. Ye, "Screen Content Coding Test Model 1 (SCM 1)," ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCTVC-Q1014, March 2014.

[10] Y. Chen and J. Xu, "BoG on 1D Dictionary," ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCTVC-R0336, June 2014.

[11] H. Yu, R. Cohen, K. Rapaka, and J. Xu, "Common Conditions for Screen Content Coding Tests," ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCTVC-Q1015, March 2014.