

# SCREEN CONTENT CODING USING NON-SQUARE INTRA BLOCK COPY FOR HEVC

Chun-Chi Chen<sup>1†</sup>, Xiaozhong Xu<sup>2‡</sup>, Ru-Ling Liao<sup>†</sup>, Wen-Hsiao Peng<sup>3†</sup>, Shan Liu<sup>4‡</sup>, and Shawmin Lei<sup>5‡</sup>

<sup>†</sup>Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan

<sup>‡</sup>MediaTek USA Inc., San Jose, CA, U.S.A.

Emails: <sup>1</sup>cheerchen.cs98g@g2.nctu.edu.tw, <sup>2</sup>xiaozhong.xu@mediatek.com, <sup>3</sup>wpeng@cs.nctu.edu.tw,

<sup>4</sup>shan.liu@mediatek.com, <sup>5</sup>shawmin.lei@mediatek.com

## ABSTRACT

To achieve high coding performance for screen content, the intra block copy (IntraBC) performs block matching within a limited area of the reconstructed samples inside the current picture. We further extend its notion from the unit of square coding unit (CU) to non-square prediction unit (PU) partitions. This design is then justified by theoretical and empirical analyses which reveal the same fact that blocks coded by IntraBC mode tend to enable more at smaller partition levels. Besides, the syntax design of the proposed method is fully aligned with that of inter partition modes. Therefore the architecture-wise change in video codec design can be minimized. The experimental results justify the effectiveness of the proposed mode for coding of screen content video. In particular, up to 19.5% rate reduction (with an average of 12.7%) relative to the HM-12.0+RExt-4.1 anchor can be achieved on top of the usage of CU-based IntraBC prediction.

**Index Terms—** *Screen Content, Intra Block Copy, Non-Square Partition, Compound Image Coding, HEVC*

## 1. INTRODUCTION

The finalization of the High Efficiency Video Coding (HEVC) standard [1] in January, 2013 is another important milestone in video coding, following its predecessor, the Advanced Video Coding (AVC) standard. It can save about 50% of the bitrate in AVC to achieve the equivalent visual quality. The HEVC standard is a generic standard, targeting to cover a wide range of applications with different contents, performance and implementation requirements.

Recently, with the rapid development of Internet and wireless communication, some of the rich media applications, such as remote desktop, wireless display and remote presentation become more and more popular. To address the ever-increasing demand of presenting high definition and high quality screen content video, such as graphics, text characters, animations, and so on, a higher-than-normal compression ratio is required. Screen content coding is therefore proposed, as an extension of the HEVC,

to further improve the coding efficiency for non-camera generated video content.

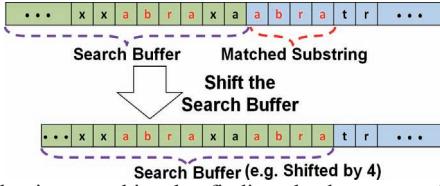
A lot of research investigations have been conducted in developing efficient screen content coding tools. Among them, the exploration of utilizing the spatial correlations of intra picture has achieved some major breakthrough. The palette mode [2] attempts to find k representatives of k clusters that can used to minimize the sum of the  $L^p$ -difference, between the intensity of each pixel to its associated cluster representative, to the power of  $p$ . The intra block copy (IntraBC) [4] reuses the notion of Sliding Window Lempel-Ziv coding, which represents a block of pixels by its length and a displacement pointing to a previous occurrence in spatially reconstructed region, to search frequently occurred patterns in the current frame. The edge mode [5] splits each transform unit, along the edge that covers the most residual energy, into two parts, resulting in non-square partitions, and then each partition is transformed independently by using a set of shape-adaptive DCT basis. Currently, the first two are the major techniques under consideration by the JCT-VC committee.

In this paper, we will focus on IntraBC prediction and propose a general frame work of partitioning the square-shaped block into some non-square blocks to perform intra IntraBC. The effectiveness of the proposed method is justified by both theoretical analysis and simulation results.

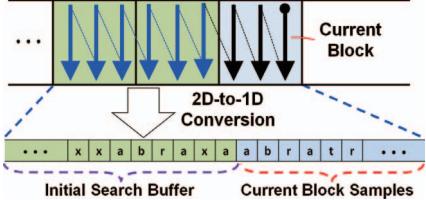
The rest of this paper is organized as follows: Section 2 gives an overview of some related works and the theoretical analysis on IntraBC is performed; based on that, the proposed method is described in Section 3; In Section 4, the simulation results are provided with some discussions; Conclusions are drawn in Section 5.

## 2. OVERVIEW OF SLIDING WINDOW LAMPEL-ZIV CODING AND ITS VARIANTS

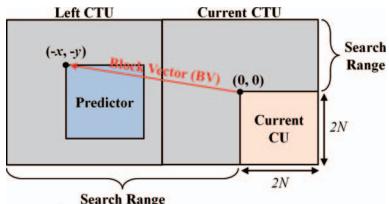
In this section, we will introduce some related works, namely the sliding window Lempel-Ziv coding and several of its variants, including the pseudo-2D-matching coder [3] and intra block copy [4]. We will also perform some analyses on intra block copy, which forms the basis of the proposed method.



**Fig. 1.** Substring matching by finding the longest perfect match anywhere from a given search buffer. Then, search buffer is updated by including the matched substring and shifting the buffer.



**Fig. 2.** Conversion of 2-D blocks of samples to a 1-D string, where the green blocks represent recently reconstructed blocks.



**Fig. 3.** The block vector search of IntraBC technique, where the gray area represents the search range of a current CU.

## 2.1. Basics of Sliding Window Lempel-Ziv Coding

The Sliding Window Lempel-Ziv coding, also known as LZ77, is a universal coding technique widely used for lossless data compression. In essence, it is a variable-length-to-variable-length coding scheme, designed with the key ideas to decompose a string into a sequence of substrings and to represent each substring by indicating its length and the displacement by which a perfect match can be found in a given search buffer. In general, the search buffer consists of the most recently decoded substrings (Fig. 1). The substring length is determined by the longest perfect match that can be found in the search buffer, for example, the “abra” in Fig. 1. If a new substring (usually a single character) is not shown in the search buffer, it is coded explicitly without compression. Upon the completion of coding a substring, it will be included as a part of the search buffer. The whole process continues for an immediately subsequent substring until every one of them in the sequence is encoded. Obviously, if certain patterns occur frequently within a string, LZ77 will perform extremely well.

## 2.2. Pseudo-2D-Matching Coder

The fact that periodic patterns occur frequently in some types of screen contents (e.g. text, graphics, tables, and etc.) enables the Pseudo-2D-Matching (P2M, a variant of LZ77)

coder [3] to code these contents efficiently. As illustrated in Fig. 2, 2-D samples within the current block and recently reconstructed blocks are first converted to 1-D string, where each character now becomes the intensity value of a sample, and the 1-D substring matching of LZ77 depicted in Fig. 1 can thus be applied.

The search criterion of P2M is the same as LZ77 (cf. Section 2.1). The major problems of P2M are twofold. First, when finding the longest match of a substring, the starting position of its immediately subsequent substring cannot be determined beforehand, resulting in an inevitably sequential encoding of substrings. Second, extra CABAC logics and a different parsing/decoding process are required to support this coding tool since the required operations are not seen in any of the modern block-based hybrid video codecs.

## 2.3. Intra Block Copy

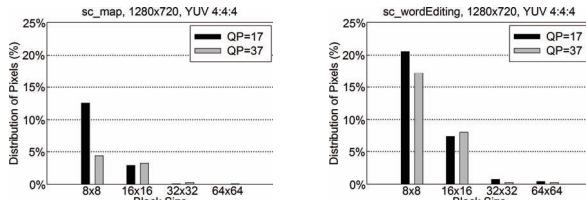
To avoid the sequential encoding of P2M, the intra block copy technique [4], referred hereafter to as IntraBC mode, requires that the length of substrings be exactly the same as the number of samples (e.g. 8x8, 16x16, and etc., denoted by using  $2Nx2N$ ) in a prediction unit. The whole process is similar to motion compensated prediction, except that the already encoded portion of the current frame is used as the search buffer for block matching. Fig. 3 illustrates the concept of this operation. The matching criterion of finding the longest perfect match within the given 1-D search buffer is replaced by searching the best reference block, in terms of rate-distortion performance, within a 2-D search buffer. Like the use of motion vector, a 2-D block vector (BV) is sent to indicate the displacement between the reference block and the current block. The prediction residuals are then encoded and signaled in the normal way.

Recently, this prediction mode is adopted in the draft HEVC Range Extension standard [6]. A flag at coding unit (CU) level is signaled to switch it on and off adaptively; moreover, to minimize the buffer usage, the search region for IntraBC is limited to the left coding tree unit (CTU) and the reconstructed region of the current CTU. As reported in [4], this mode brings promising gains, especially in coding screen content video, with BD-rate savings [7] ranging from 20% to 35% relative to the HM anchor [8].

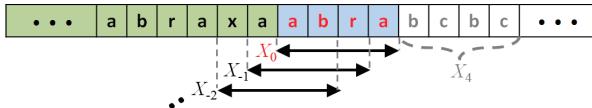
## 2.4. Analysis of Intra Block Copy

### 2.4.1. Mode Distribution of IntraBC Mode

The Fig. 4 charts the mode distribution of IntraBC mode at various block partition sizes ranging from 8x8 to 64x64. The length of each bar represents the average spatial coverage (in units of pixels) in CUs of a specific size. The bitstreams of test sequences are generated by using the HM-12.0+RExt-4.1 software [8] at two rate points (QP=17 and 37) using the All-Intra RExt configuration.



**Fig. 4.** Mode distributions of sc\_map and sc\_wordEditing sequences at QP 17 and 37 in All-Intra RExt configuration.



**Fig. 5.** Mapping functions for blocks with 4 samples.  $X_0$  and  $X_1$  represent the mapping functions of the current block and the immediately subsequent block, respectively.

From this figure, two observations can be drawn. First, the large occupation of IntraBC mode in the bitstreams provides an evidence besides the BD-rate savings to confirm the effectiveness of such prediction mode in providing better prediction efficiency with less overhead. Especially in the high rate case (QP=17), roughly 15.5% and 29.0% of the pixels in sc\_map and sc\_wordEditing sequences, respectively, are coded by using IntraBC mode. Second, the percentage of IntraBC mode drops dramatically at the both rate points when the block size increases, which implies that the IntraBC mode tends to be enabled more at smaller block levels. In the following section, we will justify this phenomenon from a theoretical viewpoint.

#### 2.4.2. Expected Recurrence Displacement

In this section, we study the IntraBC mode from a theoretical viewpoint. Our goal is to understand the underlying reason that the IntraBC mode is enabled more frequently at smaller block levels than larger ones.

Without loss of generality, we assume that the IntraBC mode operates at a certain block partition level and the block vector search only performs horizontally to the left side of the current CU. We define  $X_j$  ( $j = \dots, -2, -1, 0, 1, 2, \dots$ ) as a random variable which maps the pattern of a block (illustrated in Fig. 5) to a dedicated symbol,  $x$ , of a finite alphabet set  $S$  (e.g.  $2^{d(2N)(2N)}$  symbols for a block of size  $2Nx2N$  with  $d$ -bit samples),  $X_0$  to be the mapping function for the current block, and  $\Delta$  to be the displacement pointing to the most recent occurrence of a good match for  $X_0=x$  (if  $X_0=x$  and  $J(X_0, X_\delta)$  reaches the minimal cost among all  $\{J(X_0, X_j)\}_{j=-1, -2, \dots, -\delta, \dots}$ , then  $\Delta=-\delta$ ). For LZ77, the  $J(a, b)$  is a function indicating whether  $a$  is equal to  $b$  while for IntraBC mode, it could be either a distortion or a rate-distortion function of the two blocks associated with  $a$  and  $b$ . We assume that  $\{X_j\}_{j=\dots, -2, -1, 0, 1, 2, \dots}$  is a discrete stationary ergodic process. For any occurrence of symbol  $x$ , the conditional expected displacement of  $X_0=x$  is the reciprocal of the occurrence probability,  $Pr(X_0=x)$ , of  $x$ :

$$E[\Delta | X_0=x] = -Pr^{-1}(X_0=x). \quad (1)$$

The derivation is provided in Appendix. The result is consistent with a common intuition that it is always easier to find a frequently occurred symbol from the search buffer.

When taking the expectation on  $x$ , the expected recurrence displacement,  $E[\Delta]$ , can be expressed as follows:

$$E[\Delta] = \sum_{x \in S} Pr(X_0=x) (-Pr^{-1}(X_0=x)) = -2^{d(2N)(2N)}. \quad (2)$$

From Eq. (2), the  $|E[\Delta]|$  is the same as the size of the finite alphabet set  $S$ , which implies that stochastically, smaller blocks are expected to have shorter expected recurrence displacements. This reveals the same fact as delivered in Fig. 4 that the IntraBC mode tends to be enabled more frequently at smaller block partition levels.

## 2.5. Summary

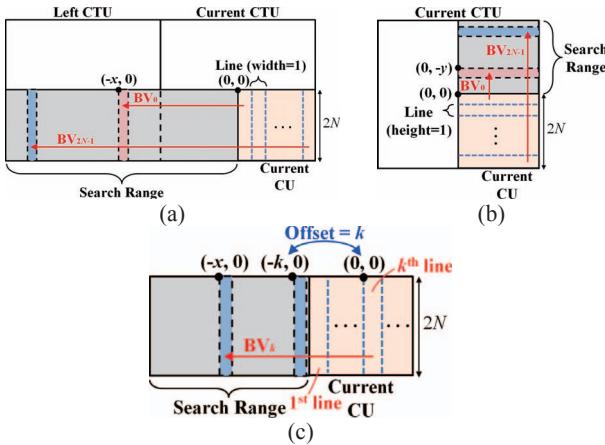
To summarize, we view the IntraBC mode as a more compliant design than that of P2M for the reason that the former avoids the sequential encoding process of substrings and remains nothing changed for residual coding in a video codec. Both the empirical and the theoretical results have proved that the expected recurrence displacements of smaller block partitions are shorter than those of larger ones. To further improve the prediction efficiency of the IntraBC mode, we will follow the observations from Section 2.4 to develop a line-based IntraBC mode. Its idea is to split a CU into smaller partitions (e.g. 1-pixel wide) and operate IntraBC at that partition level for a better intra prediction performance.

## 3. NON-SQUARE INTRA BLOCK COPY

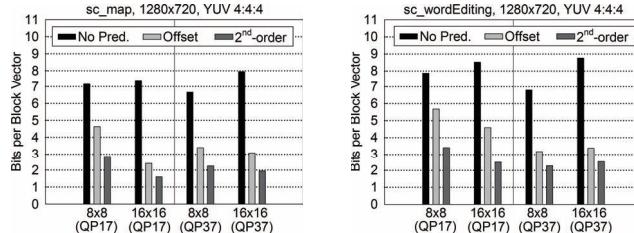
### 3.1. Framework of Line-based Intra Block Copy

The notion of the line-based IntraBC prediction is to provide an estimate of pixel intensities based on the IntraBC prediction, which relaxes the restriction that the basic unit for IntraBC must be of size  $2Nx2N$ . We introduce additional prediction units by dividing a CU equally into multiple partitions, termed as *lines*, of size  $1x2N$  or  $2Nx1$ . For each line, pixels within are predicted in the same way as the IntraBC prediction except that 1) the block size is of  $1x2N$  and  $2Nx1$  and 2) blocks of size  $1x2N$  and  $2Nx1$  are restricted to perform only the horizontal and the vertical searches, respectively. As depicted in Fig. 6, the BV search for each line is as follows:

- for each line of size  $1x2N$ , performing the horizontal BV search with a search range limited to the reconstructed regions (the gray area of Fig. 6 (a)) of the left and the current CTUs;
- for each line of size  $2Nx1$ , performing the vertical BV search with a search range limited to the reconstructed region (the gray area of Fig. 6 (b)) of the current CTU.



**Fig. 6.** (a) Horizontal BV search for lines of size  $1 \times 2N$ . (b) Vertical BV search for lines of size  $2N \times 1$ . (c) Coordinate system and block vector offsetting for  $BV_k$ .



**Fig. 7.** Bits per BV for various BV coding schemes.

Hence, there are  $2N$  one-dimensional BVs per CU needed to be searched and signaled. A fast encoding option is used to skip evaluating line-based IntraBC mode for CUs of size larger than  $16 \times 16$ . Two flags are present at CU level, one for signaling the use of the proposed mode and the other for indicating the direction of BVs.

According to the discussions in Section 2.4, operating IntraBC prediction line-by-line is expected to produce a better prediction than that produced by doing it block-by-block. However, to make the proposed mode more practical in real encoding, two key problems have to be resolved, one of which is how to signal the considerable number,  $2N$ , of BVs efficiently (e.g. 8 BVs per  $8 \times 8$  CU) and the other regarding the memory access bandwidth required for fetching a 1-pixel-wide reference block from the search buffer. We will discuss them later in the following sections.

### 3.2. Coding of Block Vectors

The coding of 1-D BVs is very similar to motion vector coding, which forms a BV predictor (BVP) for difference coding. There are two fashions proposed to generate the BVP for a line, which are the BV offsetting and 2<sup>nd</sup>-order BV prediction.

The BV offsetting forms the BVP based on the offset between the location of a line (e.g.  $k^{\text{th}}$  line in Fig. 6 (c)) and that of the closest reference block. The BVP of the  $k^{\text{th}}$  line of a CU is defined as

$$BVP'_k = -k. \quad (3)$$

The existence of this offset is because, when estimating  $BV_k$ , pixels locating at the range from the 1<sup>st</sup> to the  $(k-1)^{\text{th}}$  line are not reconstructed yet, resulting in that these BVs,  $\{(x, 0), (0, y)\}_{x,y=0, -1, \dots, -(k-1)}$ , are not allowed to be used.

The 2<sup>nd</sup>-order BV prediction further predicts the BV difference (BVD, the difference between a BV and its BVP) by using the BVD from an immediately previous line.

$$BVP_k = BVP'_k + (BV_{k-1} - BVP'_{k-1}) = BV_{k-1} - 1. \quad (4)$$

The BVD generated by using either BV prediction (Eq. (3) or (4)) is then encoded in the same way as coding of motion vector difference. The Fig. 7 charts the average bit overhead required for coding BVs. The test condition in use is the same as Section 2.4.1. Significant reduction of bit overhead confirms the effectiveness of the BV prediction schemes. As comparing the 2<sup>nd</sup>-order BV prediction with the BV offsetting, the former can further reduce roughly 1-5 bits per BV (31-68% of rate saving). Accordingly, we adopt Eq. (4) to the proposed mode for later experiments.

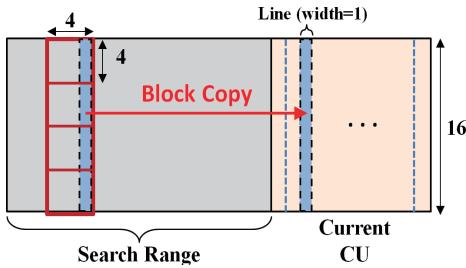
### 3.3. Memory Access Bandwidth

Obviously, the line-based IntraBC offers a greater flexibility in forming a predictor, but it is extremely complex, especially when the issue of memory access bandwidth involves. In some hardware designs, fetching a column of pixels is really no different from fetching a column of blocks. An example given in Fig. 8 shows that if the minimum data access granularity is a  $4 \times 4$  block, then, to get one column of pixels, you will have to fetch the whole column of  $4 \times 4$  blocks. This means a dramatic increase in memory access bandwidth when a CU is predicted line by line. To improve the effective data access rate, the line-based IntraBC can be generalized to include more rows or more columns (e.g. generally a multiple of 4) in a line.

### 3.4. Non-square Intra Block Copy

The notion of line-based IntraBC is extended to a more generalized form, namely *non-square intra block copy*, which represents a CU by a multiple of equally partitioned prediction units of size  $M \times 2N$  or  $2N \times M$ , where  $M$  represents a power-of-two integer and is smaller than  $2N$ . Each unit of a CU is processed in the same way as a line, that is, performing IntraBC prediction and sending the BVs at  $M \times 2N$  and  $2N \times M$  partition levels.

Since the setting,  $M=N$ , corresponds to  $2N \times N$  and  $N \times 2N$  PUs, which are already supported in the current HEVC, no doubt this video codec should support a minimum data access granularity with respect to  $N \times 2N$  and  $2N \times N$  partitions. For the trade-off between bandwidth and performance, it is worth considering applying IntraBC to such PU's.



**Fig. 8.** Smallest granularity (marked in red square) to fetch reference samples for line-based IntraBC prediction.

**Table 1.** Semantics of partition size for inter modes.

| Partition Size | Bins    |                     |                 |
|----------------|---------|---------------------|-----------------|
|                | (a)     | (b)                 | (c)             |
|                | Non-SCU | SCU<br>(if not 8x8) | SCU<br>(if 8x8) |
| 2Nx2N          | 1       | 1                   | 1               |
| 2NxN           | 01      | 01                  | 01              |
| Nx2N           | 00      | 001                 | 00              |
| NxN            | -       | 000                 | -               |

**Table 2.** Y BD-rate savings and runtimes for non-square IntraBC.

| Configuration                 | Class F | SC YUV | SC RGB | Enc. | Dec. |
|-------------------------------|---------|--------|--------|------|------|
| <i>BV Prediction Enabled</i>  |         |        |        |      |      |
| 1x2N/2Nx1                     | -4.8    | -10.3  | -12.7  | 114% | 97%  |
| 4x2N/2Nx4                     | -2.3    | -7.0   | -7.1   | 112% | 94%  |
| Nx2N/2NxN                     | -2.6    | -6.8   | -6.9   | 113% | 94%  |
| <i>BV Prediction Disabled</i> |         |        |        |      |      |
| 1x2N/2Nx1                     | -3.0    | -7.5   | -10.6  | 114% | 97%  |
| 4x2N/2Nx4                     | -2.2    | -6.8   | -7.0   | 112% | 94%  |
| Nx2N/2NxN                     | -2.5    | -6.7   | -6.9   | 113% | 94%  |

Besides, to avoid complicating the CABAC logics, the signaling of partition size is aligned, to the utmost, with that of inter prediction modes. The semantics of partition mode for this mode is now the same as that listed in Table 1 (a) for both the smallest CU (SCU) and non-SCUs.

### 3.5. Summary

To summarize, the line-based IntraBC is proposed as an extension of operating IntraBC mode at a finer partition level with the intention to find a compromise between the prediction accuracy and the required BV overhead. A simple yet efficient BV coding and the extension to  $2NxN/Nx2N$  partitions make this mode more practical in a real codec. To see how this tradeoff would impact the R-D performance, we will next present simulation results based on the HM-12.0+RExt-4.1 software [8].

## 4. EXPERIMENTAL RESULTS

Experiments provided in this section were conducted using the HM-12.0+RExt-4.1 software [8] and the RCE3 common test conditions [10] at 4 QP values (22, 27, 32, 37) to collect the BD-rate and the encoding and decoding time

for comparison. For a fair comparison, the size of search buffer for both P2M and IntraBC modes is limited to the current and the left CTUs. If not otherwise stated, the All Intra RExt configuration is used for producing coding results.

### 4.1. Coding Results of Non-square Intra Block Copy

Three different configurations with  $M=1$ , 4, and  $N$  are tested one at a time, along with the original  $2Nx2N$  IntraBC mode. From Table 2, the coding gains are found mostly in SC sequences. As expected, the line-based IntraBC consistently outperforms the other two configurations ( $M=4$ ,  $N$ ). The BV prediction brings additional BD-rate reduction of 2-3% while its benefit to the other two is less obvious. Besides, it can be seen that  $M=4$  and  $M=N$  perform almost identically.

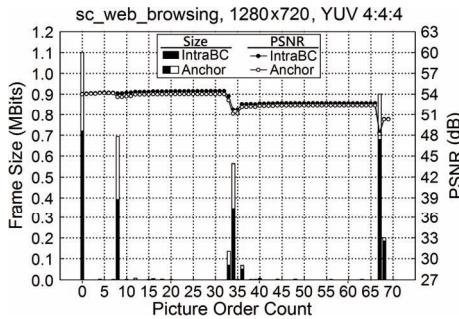
The runtime characteristics of the three configurations are quite similar. This is because almost the same number of comparisons is performed regardless of the number of pixels within a partition or a line. Reportedly, the encoding time is increased by 12-14%, but for some reason, the decoder runs a bit faster.

### 4.2. Combination with NxN Intra Block Copy

Syntax alignment with already existing prediction modes is, for most of the time, the best policy when a new mode is to be introduced into a video codec. We incorporate  $NxN$  PU with the non-square IntraBC with the intention to unify the signaling of partition size to inter modes (cf. Table 1) even though their coding performances are nearly the same (cf. Table 3). In addition, because IntraBC mode enables more for smaller blocks, we leave the codec the flexibility to evaluate  $NxN$  PU for IntraBC at 8x8 SCU level, resulting in that the syntax structure in Table 1 (c) is replaced by (b) and  $NxN$  PU is additionally applicable at 8x8 SCU level. Although the additional gains (0.3-2.0%) the  $NxN$  IntraBC brings on top of its non-square sibling are rather limited, its major merit is that the architecture-wise change, especially the CABAC logics, can be minimized to a great extent.

### 4.3. Coding Results with Inter Prediction

Table 4 provides the coding results when inter prediction involves in the coding of screen content. Roughly 5% of BD-rate reduction on average is observed from SC sequences. Fig. 9 reveals the fact why the IntraBC modes still work when inter prediction involves. From the figure, a significant reduction of frame size can be observed at a few frames which, surprisingly, dominate almost all (more specifically, 98%) of the rate overhead required for encoding the entire video sequence. We found that the scenes are changed dramatically at those frames and are



**Fig. 9.** Frame size and PSNR of the first 70 frames of sc\_web\_browsing (QP=17, Low Delay B configuration). The most obvious peaks correspond to Frame #0,8,33,34,36,67,68.

**Table 3.** Y BD-rate savings and runtimes for the combination of the non-square IntraBC and NxN IntraBC, P2M, and Palette.

| Configuration | Class F | SC YUV | SC RGB | Enc. | Dec. |
|---------------|---------|--------|--------|------|------|
| Nx2N/2NxN     | -2.6    | -6.8   | -6.9   | 113% | 94%  |
| NxN           | -1.9    | -6.8   | -7.8   | 109% | 98%  |
| Comb. NxN     | -2.9    | -8.3   | -8.9   | 118% | 93%  |
| P2M [3]       | 0.0     | -3.5   | -9.3   | 102% | 96%  |
| Palette [2]   | -2.7    | -7.5   | -15.0  | 100% | 98%  |

very different from their reference frames, resulting in an inefficient inter prediction.

#### 4.4. Comparison with P2M and Palette Mode

This section provides results for comparing P2M and palette mode with the combined non-square and NxN IntraBC mode (cf. Table 3). Although the P2M and palette mode outperform the combination of non-square and NxN IntraBC modes in some cases, as considering that most of the gains are already obtained and fewer changes are introduced to the syntax and CABAC logics, we view the proposed IntraBC as a more feasible design for coding screen content video than the others.

## 5. CONCLUSION

In this paper, besides the original 2Nx2N IntraBC, a more general 2NxM/Mx2N IntraBC mode is proposed and discussed for coding of screen content video. In particular, when  $M=N$ , the non-square IntraBC can achieve significant improvement in terms of coding performance, while the changes in both syntax/architectural design and memory access bandwidth is still moderate.

## 6. APPENDIX

This appendix provides derivation details of Eq. (1) with referring to Kac's lemma [9]. We assume that an entire frame is split equally into  $n$ -sample blocks. We define the probability,  $Pr(\{\Delta=-\delta, X_{nj}=x\})$ , for all  $j=0,1,2,\dots$ , for the event of finding a feasible match for  $X_{nj}=x$  at the position  $-\delta$ . Because the associated blocks of  $X_k=x$ , for all  $k>0$  and  $k\neq nj$ ,

**Table 4.** Y BD-rate savings and runtimes for Non-square IntraBC with inter prediction enabled.

| Configuration | Class F | SC YUV | SC RGB | Enc. | Dec. |
|---------------|---------|--------|--------|------|------|
| Low Delay B   | -1.4    | -4.9   | -4.1   | 102% | 101% |
| Random Access | -2.0    | -5.8   | -5.9   | 106% | 99%  |

are not aligned with the assumed block partition, these events  $\{\Delta=-\delta | X_k=x\}_{k>0, k\neq nj}$  will never happen. The probability,  $Pr(\{\Delta=-\delta | X_k=x\})$ , for all  $k>0$  and  $k\neq nj$ , is thus viewed as zero. Since these events,  $\{\Delta=-\delta, X_{nj}=x\}_{j=0,1,2,\dots}$ , are disjoint, the sum of their probabilities should be equal to 1 as follows:

$$\begin{aligned} 1 &= \sum_{\delta=1}^{\infty} \sum_{j=0}^{\infty} Pr(\{\Delta=-\delta, X_{nj}=x\}) \\ &= \sum_{\delta=1}^{\infty} \sum_{i=0}^{\infty} Pr(\{\Delta=-\delta, X_i=x\}) \dots (Pr(\{\Delta=-\delta | X_k=x\})=0, k\neq nj) \\ &= Pr(X_0=x) \sum_{\delta=1}^{\infty} \sum_{i=0}^{\infty} Pr(\{\Delta=-\delta-i | X_0=x\}) \dots (\text{Stationarity}) \\ &= Pr(X_0=x) \sum_{k=1}^{\infty} k Pr(\{\Delta=-k | X_0=x\}) \\ &= Pr(X_0=x) (-E[\Delta | X_0=x]). \end{aligned}$$

## 7. REFERENCES

- [1] G. J. Sullivan, J.-R. Ohm, W.-J. Han and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” IEEE Trans. on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.
- [2] X. Guo, B. Li, J.-Z. Xu, Y. Lu and S. Li, “AHG8: Major-color-based screen content coding,” ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-O0182, Oct, 2013.
- [3] 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, 2013.
- [4] C. Pang, J. Sole, L. Guo, M. Karczewicz and R. Joshi, “Non-RCE3: Intra motion compensation with 2-D MVs,” ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-N0256, July 2013.
- [5] S. Hu, R. A. Cohen, A. Vetro, C.-C. J. Kuo, “Screen content coding for HEVC using edge modes,” IEEE Int'l Conf. on Acoustics, Speech and Signal Processing, 2013.
- [6] D. Flynn, J. Sole and T. Suzuki, “Range extensions draft 4,” ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-N1005, July, 2013.
- [7] G. Bjontegaard, “Improvements of the BD-PSNR Model,” ITU-T SG16 WP3, VCEG-AI11, 2008.
- [8] A. Saxena, D. Kwon, M. Naccari and C. Pang, “HEVC range extensions core experiment 3 (RCE3): Intra prediction techniques,” ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-N1123, 2013.
- [9] T. M. Cover and J. A. Thomas, “Elements of information theory,” Wiley, 2nd edition, 2006.