Efficient Hash Generation Method for Intra Block Copy Search in HEVC Screen Content Coding

Ilseung Kim
Department of Electronics and Computer Engineering
Hanyang University
Seoul, Korea
e-mail: ghanjang@gmail.com

Jechang Jeong
Department of Electronics and Computer Engineering
Hanyang University
Seoul, Korea
e-mail: jjeong@hanyang.ac.kr

***Abstract*— Recently, the standardization of HEVC-SCC was finalized, including a hash-based intra block copy (IBC) search scheme which brings significant coding gains to code screen content. However, the hash table generation itself creates a massive computational burden on the encoder side. In this paper, we propose a picture-wise hash generation method to reduce significantly the complexity of hash-based IBC search. Experimental results show that the proposed scheme reduces about 77% of hash generating time, which leads to 43% total IBC searching time saving in all-intra (AI) coding structures, compared with the hash-based IBC search in the HEVC-SCC test model (SCM)-8.0.**

Keywords-HEVC; Screen Content Coding (SCC); Hash; Intra Block Copy (IBC).

I. INTRODUCTION

High Efficiency Video Coding standard (HEVC) [1] is the state-of-the-art video coding standard and it was finalized in January 2013. HEVC doubles the coding efficiency of video compression compared with H.264/MPEG-4 AVC standard under the equivalent subjective visual quality circumstances [2][3].

Recently, on the other hand, computer generated content, called screen content, has surged with the rapid development of multimedia techniques and applications, such as wireless display, remote desktop, external display interfacing, and cloud computing, etc. [4]. Requests have been made in asking for investigation of new coding tools for screen content [4]. In order to reflect these requests, the MPEG requirements subgroup published a set of requirements for an extension of HEVC for coding of screen content in January 2014 [4] and currently the HEVC Screen Content Coding (SCC) Draft 6 was published in February 2016 [5][6].

Different from the traditional camera-captured content, screen content often has mixed content consisting of rendered graphics, text, or animation as well as camera-captured content in one picture. In addition, as shown in Fig. 1, screen content has different characteristics compared with that of the camera-captured content, such as containing no sensor noise, having large uniformly flat areas, repeated patterns, and a limited number of different colors and so on.

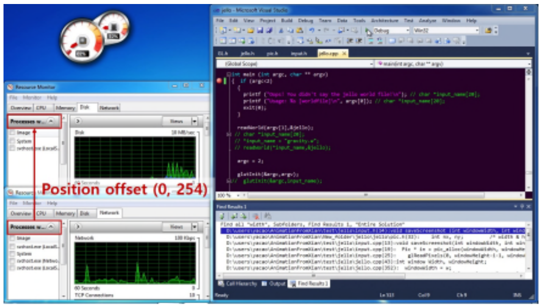


Figure 1. Example of the long-distance repeated patterns of the sequence 'sc-programming'.

In order to improve the coding efficiency for screen content doing, HEVC-SCC has developed many new tools. Palette mode utilizes the observation that a limited number of different color values frequently exist for screen content [7]. A lot of HEVC-SCC test materials consist of RGB colour format or YCbCr 4:4:4 format, whose inter-colour component correlation is very high. In order to remove inter-colour component redundancy, adaptive color transform (ACT) has been introduced [8]. Unlike camera- captured content, there is no need to use fractional motion compensation for much screen content. For this reason, adaptive motion vector resolution (AMVR) has adopted in SCC [9]. There are many repeated patterns, such as characters in screen content, so the motion estimation and compensation within the current picture can be effective. Intra block copy (IBC) can get a good prediction from the block with the same pattern in the coded region on the current frame [10].

Long range repeated patterns and large motions are often observed in screen contents as shown in Fig. 1. In order to support large range block search without bringing heavy computation burden, the hash-based global IBC search was adopted in HEVC-SCC draft 6 [5]. To perform the hash-based IBC search, hash tables of the reconstructed coding tree unit (CTU) are generated first. The hash value as well as the block position of each block are calculated and stored in the hash table. The blocks sharing the same hash values with the current coding block will be selected as candidates and rate distortion optimization (RDO) based block matching is conducted with the candidates. Detailed process of the hash-based IBC search will be introduced in Section II. However, hash table generation itself still has heavy computation burden. In this paper, we present an efficient way to generate hash values for hash-based IBC search in HEVC-SCC.

This paper is organized as follows: Section II provides an overview of technical issues of IBC and presents a tool for improving IBC, especially hash-based block vector search as a conventional algorithm. Section III presents the proposed algorithm. Section IV discusses the experiment results and the conclusion is set forth in Section V.

II. INTRA BLOCK COPY (IBC) IN HEVC-SCC

Basically, IBC is one of the block matching techniques with the reconstructed current picture before loop filtering. In other words, IBC is the technique that searches the block vectors (BVs) in terms of the lowest rate-distortion (RD) cost within the reconstructed current picture. When calculating RD cost, chroma components are used as well as luma components. First, find the four best BVs by calculating RD cost of luma components and it can be expressed as follows:

$$\mathrm{RD_{cost}} = SAD_{luma} + \lambda \times BV_{bits}. \tag{1}$$

where SAD_{luma} is sum of absolute differences of luma components, λ is a Lagrangian multiplier, and BV_{bits} is the number of bits needed to signal the BV, respectively. For the chosen four BVs, additional RD calculation process, called the chroma refinement process, is performed in order to find the optimal BVs and it can be expressed as followed:

$$\mathrm{RD_{cost}} = SAD_{luma} + SAD_{chroma} + \lambda \times BV_{bits}. \tag{2}$$

where SAD_{chroma} is sum of absolute differences of chroma components. The SAD of two N x N blocks is defined as

$$SAD = \sum_{i,j=0}^{N} |f_c(i,j) - f_r(i-x, j-y)|. \tag{3}$$

where $f_c(i,j)$and $f_r(i,j)$denote the intensity of the pixel with coordinate (i, j) in the current picture and the reference picture, (x, y) represents the displacement of the BVs, respectively.

As aforementioned, local and global search methods are introduced in HEVC-SCC. Local searches are performed first and global searches are followed. Finally, decide the optimal BVs by comparing both searches in terms of RD cost. For the local search, only two CTUs, the current and the left CTU, are used as a search range as depicted in Fig. 2.

Global search is conducted only for 16x16 and 8x8 prediction unit (PU) and the entire reconstructed current picture before loop filtering can be a search range for global search. For 16x16 PU, a one-dimensional search method is used, which can search only one direction, horizontally or vertically as shown in Fig.3.

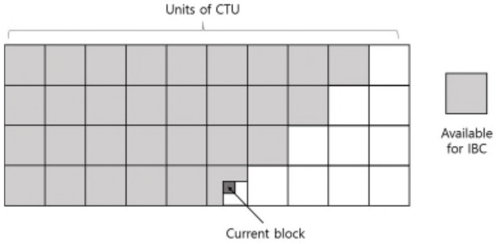


Figure 2. Global block vector search prediction area

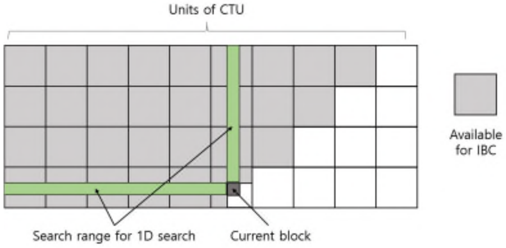


Figure 3. A one-dimensional search prediction area for 16x16 blocks

For 8x8 PU, a hash-based search is used for the entire picture but available. The 16-bit hash entries for the current block and the reference block are calculated using the original sample values. Let *Grad* denote the gradient of the 8x8 blocks and *dc0*, *dc1*, *dc2*, and *dc3* denote the DC values of the four 4x4 sub-blocks of the 8x8 block. Then, the 16-bit hash entry *H* is calculated as:

$$\begin{aligned} H &= MSB(dc0,3) \ll 13 + MSB(dc1,3) \ll 10 \\ &+ MSB(dc2,3) \ll 7 + MSB(dc3,3) \ll 4 \\ &+ MSB(Grad,4) \end{aligned} \tag{5}$$

where MSB(X,n) represents the n most significant bits of X.

The procedure of hash-based IBC search is as follows. The values of H are calculated when the CTUs were reconstructed and they are stored in the hash list, unless the value of $Grad$ is less than 5. After calculating the value of H of the current PU, perform the RD process of the luma component using (1) for the blocks which have the same hash value with that of the current PU and choose the eight best BVs. For the chosen eight best BVs, additional RD cost is calculated using (2). By doing so, the optimal BV of the global search can be determined.

III. PROPOSED ALGORITHM

Hash-based IBC mode has a critical role in HEVC-SCC. Compared with SCC without hash-based IBC, up to 75% Bjøntegaard Distortion bitrate (BD-BR) gain can be achieved in our experiment. However, hash generation and hash table updating method in HEVC-SCC brings a lot of computational redundancy.

In HEVC-SCC, the number of search points N_{search_point} required to calculate hash values are

TABLE I. THE NUMBER OF OPERATIONS TO OBTAIN A HASH VALUE H PER BLOCK IN HEVC-SCC

	Number of operations	if $N = 8$
Addition	$4(N-1)(N-1)$	196
Shift	$(N-1)(N-1)+4$	53
Absolute	$2(N-1)(N-1)$	98

TABLE II. THE NUMBER OF OPERATIONS TO OBTAIN A HASH VALUE H PER BLOCK IN THE PROPOSED ALGORITHM

	Number of operations (if $N = 8$)
Addition	$\cong 15$
Shift	$\cong 1+4$
Absolute	$\cong 2$

$$N_{search\_point} = O\{(W_{pic}-N)(H_{pic}-N) + \left(\frac{W_{pic}H_{pic}}{N^2}\right)\} \qquad (6)$$

where W_{pic} and H_{pic} are the picture width and height, N denotes block size, and $O\{\cdot\}$ represents a big-O notation, respectively. Moreover, $(N-1)^2$ addition operations and 4 shift operations for computing dc values, and $2(N-1)^2$ absolute operations, $(N-1)^2$ shift operations and $3(N-1)^2$ for calculating *Grad* value are required to calculate the hash value *H*. The number of operations required to obtain a hash value *H* per block is summarized in Table I.

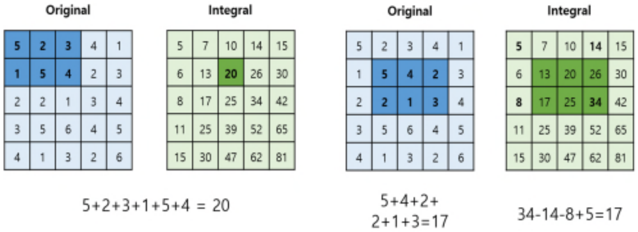
However, the number of required operations to compute hash entry H is always same at every search point without considering the correlation among them, even though many operational overlaps occur between neighboring blocks. Moreover, updating the hash table right after the reconstruction of each CTU also causes unnecessary duplication of calculations. Hash entry *H* is calculated using the original sample values, not reconstructed values; thus, there is no need to update hash table after the reconstruction of CTUs.

In this paper, we propose a picture-wise hash generation method using integral image to effectively eliminate unnecessary operations for hash generation and hash table updating in HEVC-SCC.

Integral image is widely used to calculate the sum norm values, as shown in Fig. 4. The required operations for the value of *dc*s and *Grad* in eq. 4 to obtain the hash value *H* are closely related to the way to compute the sum-norm of the relevant block. Fig. 5 shows examples of required operations to obtain the sum norm of two consecutive $N \times N$ blocks. Compared to the method in HEVC-SCC which requires $(N-1)^2$ additional operations to calculate the sum norm value of each $N \times N$ block, only three addition operations are needed by using an integral image.

To calculate the integral image, the following number of addition operations $N_{addition}$ are required

$$N_{addition} = 3W_{pic}H_{pic} - 2W_{pic} - 2H_{pic} + 1. \qquad (7)$$



5	2	3	4	1
1	5	4	2	3
2	2	1	3	4
3	5	6	4	5
4	1	3	2	6

5	7	10	14	15
6	13	20	26	30
8	17	25	34	42
11	25	39	52	65
15	30	47	62	81

5	2	3	4	1
1	5	4	2	3
2	2	1	3	4
3	5	6	4	5
4	1	3	2	6

5	7	10	14	15
6	13	20	26	30
8	17	25	34	42
11	25	39	52	65
15	30	47	62	81

Figure 4. Example of an integral image.

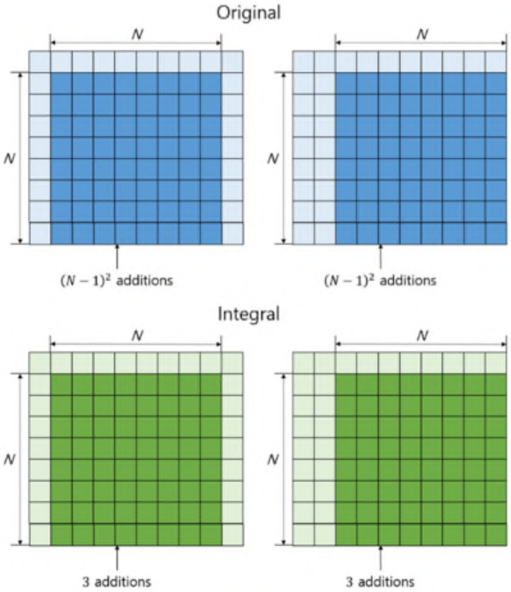


Figure 5. Examples of required operations to obtain the sum norm of two consecutive $N \times N$ blocks

Considering eq. 6, the number of operations required to make an integral image per search point are approximately 3 additions.

The procedure of the proposed algorithm is as follows: first, create the gradient map for the current picture. In this stage, $3(W_{pic}-1)(H_{pic}-1)$, $2(W_{pic}-1)(H_{pic}-1)$, and $(W_{pic}-1)(H_{pic}-1)$ of addition, absolute, and shift operations are required in a picture unit, respectively. Then, the remaining operations needed to acquire hash entry H are the same as those used to calculate dc values in eq.4, which are highly related to calculate the sum norm value of the relevant block. Generating an integral image of the gradient map for the current picture is followed by. Using the integral image of the gradient, the value of *Grad* in (4) can be calculate and then, stop hash generation process if the value of *Grad* is less than 5. After that, make an integral image of the current picture and calculate the DC values of the four 4x4 sub-blocks using the integral image of the current picture. By doing so, the hash values of all search points in the current picture can be efficiently generated. The number of operations in the proposed algorithm is summarized in Table II and it shows the amount of the decreasing number of operations compared to that in HEVC-SCC as shown in Table II.

IV. EXPERIMENTAL RESULTS

The simulation was conducted in 4:4:4 test sequences under the common test conditions (CTC) used during the development of HEVC-SCC [6] under the intra main SCC configuration. In order to test the screen content in a variety of environments, HEVC-SCC CTC includes several types of sequences such as text and graphics with motion (TGM), mixed content (M), animation (A), and camera-captured content (CC). HM-16.8+SCM-8.0 was used as the anchor, and 22, 27, 32, and 37 are set for QPs.

Table III and IV showed the time comparison result for the hash generation and total IBC searches to verify computational complexity of the proposed algorithm. The total IBC searching time consisted of the hash generation time and the hash-based search time including RDO process for the candidates which matched the hash value with the that of the current block. In order to compare the consumed time, time rate was used with the following equation:

$$Time\_rate = \frac{time_{proposed}}{time_{anchor}} \times 100\%. \qquad (6)$$

The results in Table III showed that the same results can be achieved by taking 21.83% on average of time compared to anchor when generating IBC hashes. Using the method in the proposed algorithm, the total IBC search time was up to 24.38% and 56.92% on average compared to that of anchor. Note that coding efficiency between the anchor and the proposed algorithm is equivalent, because every methods except hash generation method between them are same.

V. CONCLUSION

This paper proposes an efficient hash generation method for IBC search after analyzing the conventional IBC technique in HEVC-SCC draft 6, especially hash-based IBC. The proposed algorithm suggests an efficient way to generate hash values in picture units at once by using integral images and the early termination method. The experimental result shows that the hash generation time rate is approximately 23%, which leads to 43% total IBC searching time saving in all-intra (AI) coding structures, compared with the hash-based IBC search in the HEVC-SCC test model (SCM)-8.0.

TABLE III. HASH GENERATION TIME COMPARISON BETWEEN HM-16.8+SCM-8.0 AND THE PROPOSED ALGORITHM.

Category	Sequences	Time rate (%)
TGM, 1080p	sc_flyingGraphics	17.68
	sc_desktop	23.60
	sc_console	25.35
TGM, 720p	sc_web_browsing	21.82
	sc_map	22.71
	sc_programming	21.86
	sc_SlideShow	27.98
M, 1440p	Basketball Screen	21.83
	MissionControlClip2	23.50
M, 1080p	MissionControlClip3	21.72
A, 720p	sc_robot	24.00
CC, 1080p	Kimono1	24.88
Average		**23.08**

TABLE IV. TOTAL IBC SEARCHING TIME COMPARISON BETWEEN HM-16.8+SCM-8.0 AND THE PROPOSED ALGORITHM.

Category	Sequences	Time rate (%)
TGM, 1080p	sc_flyingGraphics	80.85
	sc_desktop	80.76
	sc_console	80.84
TGM, 720p	sc_web_browsing	50.52
	sc_map	58.21
	sc_programming	34.71
	sc_SlideShow	67.85
M, 1440p	Basketball Screen	24.38
	MissionControlClip2	57.72
M, 1080p	MissionControlClip3	54.40
A, 720p	sc_robot	64.84
CC, 1080p	Kimono1	27.93
Average		**56.92**

ACKNOWLEDGMENT

“This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT and future Planning(NRF-2015R1A2A2A01006004)“

REFERENCES

[1] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012

[2] J. Ohm, G. J. Sullivan, H. Schwarz, T. Tan, and T. Wiegand ” Comparison of the coding efficiency of video coding standards ‐ including High Efficiency Video Coding (HEVC),” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669‐1684, Dec. 2012.

[3] Advanced Video Coding for Generic Audiovisual Services, ITU-T and ISO/IEC JTC1, document ITU-T Rec. H.264 and ISO/IEC 14496-10, May 2003.

[4] H. Yu, K. McCann, R. Cohen, and P. Amon, Requirements for an Extension of HEVC for Coding of Screen Content, ISO/IEC JTC 1/SC 29/WG 11, document MPEG2014/N14174, San Jose, CA, USA, Jan. 2014.

[5] R. Joshi. et. al., “HEVC Screen Content Coding Draft Text 6,” 23rd JCT-VC meeting, San Diego, U.S. document JCTVC-W1005, Feb. 2016.

[6] R. Joshi, J. Xu, R. Cohen, S. Liu, and Y. Ye, “Screen content coding test model 6 (SCM 6),” 22nd JCT-VC meeting, Geneva, Switzerland, document JCTVC-V1014, Oct. 2015.

[7] Z. Ma, X. Feng, and M. Xu, “Advanced screen content coding using color table and index map,” *Image Processing, IEEE Transactions on*, vol. 23, no. 10, pp. 4399–4412, 2014.

[8] P. Lai, S. Liu, and S. Lei, "Ahg6: On adaptive color transform (act) in scm2.0," document JCTVC-S0100, JCT-VC, Strasbourg, France, Oct. 2014.

[9] B. Li, J. Xu, G. J. Sullivan, Y. Zhou, and B. Lin, "Adaptive motion vector resolution for screen content," document JCTVC-S0085, JCTVC, Strasbourg, France, Oct. 2014.

[10] C. Pang, J. Sole, Y. Chen, V. Seregin, and M. Karczewicz, "Intra block copy for hevc screen content coding," *in Data Compression Conference (DCC)*, 2015, 2015, p. 465.

ISBN: 978-1-61208-577-7