Chapter 1

Differential Pixel Value Coding for HEVC Lossless Compression

Jung-Ah Choi and Yo-Sung Ho

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/52878

1. Introduction

High efficiency video coding (HEVC) [1] is a new video coding standard developed by Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG). Currently, most of coding techniques are established and HEVC version 1 will be released in January 2013 [2]. We expect that HEVC is widely used in various applications for recording, compression, and distribution of high-resolution video contents [3].

Lossless compression is useful when it is necessary to minimize the storage space or transmission bandwidth of data while still maintaining archival quality. Many applications such as medical imaging, preservation of artwork, image archiving, remote sensing, and image analysis require the use of lossless compression, since these applications cannot allow any distortion in the reconstructed images [4].

With growing demand for these applications, JCT-VC included the lossless coding mode in the HEVC test model (HM) software in consequence of the Ad Hoc group for lossless coding [5]. In lossless coding, no distortion is allowed in reconstructed frames. To achieve lossless coding, transform, quantization, their inverse operations, and all in-loop filtering operations including deblocking filter, sample adaptive offset (SAO), and adaptive loop filter (ALF) are bypassed in the encoder and decoder since they are not reversible in general [6]. Also, sample-based angular prediction (SAP) [7][8] is used to replace the existing intra prediction method.

In the 7th JCT-VC meeting, many lossless coding solutions were proposed. Mode dependent residual scanning (MDRS) and multiple scanning positions for inter coding are suggested [9]. Also, SAP and lossless transforms [10] are proposed. Among these proposals, SAP is adopted in the HEVC standard. In the next 8th JCT-VC meeting, efforts to find the efficient lossless coding solutions continued. Joint proposal that combines SAP and the lossless coding signaling method was submitted [5] and a simplified context-based adaptive binary arithmetic coding (CABAC) structure without last position coding [11] was introduced. Since the development of the HEVC lossless mode is not yet finished, many experts are actively researching efficient algorithms for lossless coding [12][13].

In this chapter, we have tried to design an efficient differential pixel coding method for the HEVC lossless mode. One caution in developing the HEVC lossless mode is that the coding performance of the HEVC lossy mode would not be impacted or compromised. In lossless coding, the residual data is not quantized transform coefficients but differential pixel after prediction. As a result, the residual data in lossless coding has different characteristics than that in lossy coding. Thus, we analyze characteristics of the residual data in lossless coding and propose efficient mode dependent differential pixel scanning and entropy coding using the modified binarization. Note that the proposed method does not require any modification of syntax elements in HEVC, so it can be easily applied to the current standard. Moreover, the amount of complexity increase is negligible.

The chapter is organized as follows. In Section 2, we briefly present an overview of the HEVC lossless mode including its structure, SAP, scanning, and entropy coding. In Section 3, after we analyze characteristics of residual data in lossless coding, the proposed method for differential pixel value coding is explained. In Section 4, the performance of the proposed method is compared to the performance of the HEVC lossless mode in terms of bit saving and complexity. Finally, conclusions are presented in Section 5.

2. Overview of the HEVC lossless mode

The basic approach for lossless coding is to bypass transform and quantization in the encoder and the decoder. Without transform and quantization, SAP can be incorporated to improve coding efficiency of the lossless mode. It replaces the general angular intra prediction method in the HEVC lossy mode.

When the lossless mode is applied, all the in-loop filtering operations including deblocking filter, SAO, and ALF are also bypassed. Since there is no distortion existing in the reconstructed frame in the lossless mode, in-loop filtering operations will not help either picture quality or coding efficiency. The overall structure of the HEVC lossless mode is shown in Figure 1. In Figure 1, dashed lines represent the bypass and all bypass operations are activated in the HEVC lossless mode. Main coding modules are explained in detail in following sub-sections.

2.1. Sample-based angular prediction

In order to explore spatial sample redundancy in intra-coded frame, SAP is employed instead of general HEVC intra prediction. As shown in Figure 2, 33 angles are defined and these angles are categorized into two classes: vertical and horizontal angular prediction. Each prediction has both negative and positive angles.

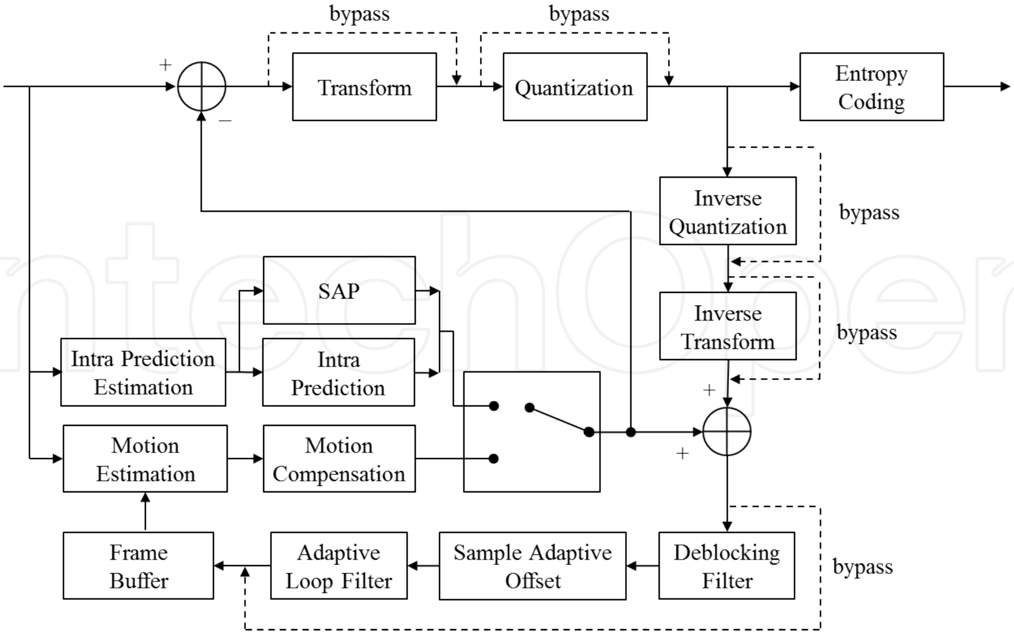


Figure 1. Encoder structure of the HEVC lossless mode

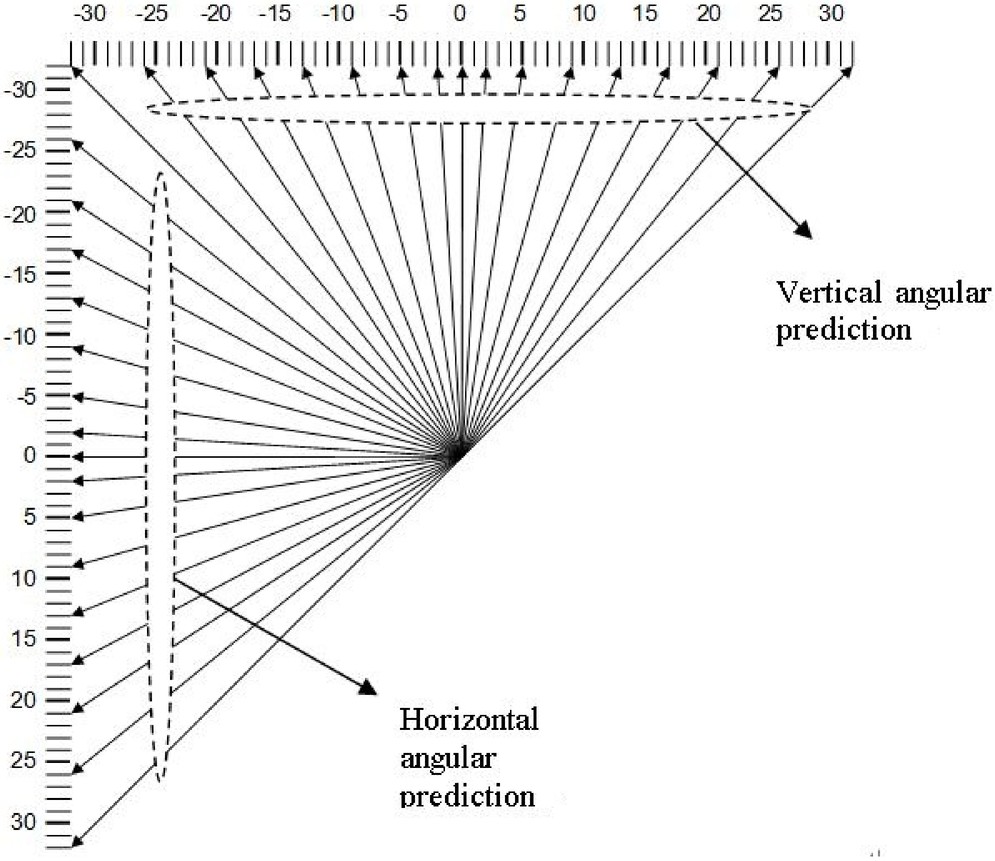
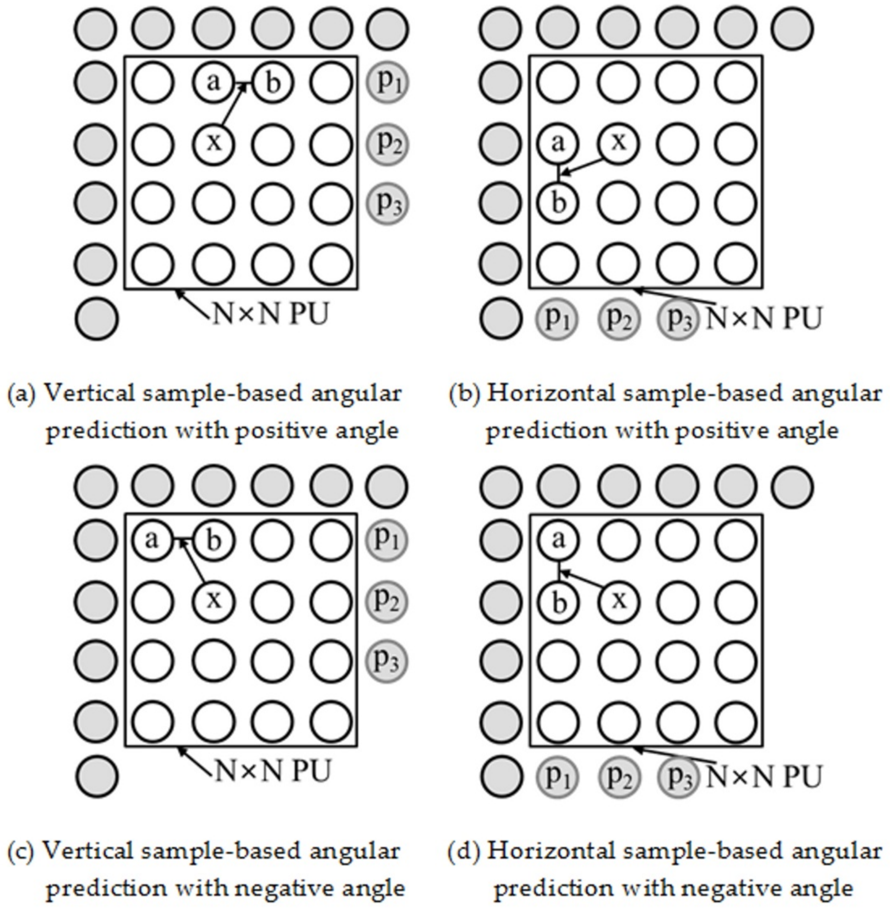


Figure 2. Intra prediction angles (vertical and horizontal angular prediction)

In lossless coding, reference samples within the current prediction unit (PU) as well as neighboring samples of the current PU are available. Thus, prediction can be performed sample by sample to achieve better intra prediction accuracy. All samples within a PU use a same prediction angle and the signaling method of the prediction angle is exactly same as that in lossy intra coding.

In SAP, samples in a PU are processed in pre-defined orders. The raster scanning and vertical scanning processing order is applied to vertical and horizontal angular prediction, respectively. In addition, reference samples around right and bottom PU boundaries of the current PU are padded from the closest boundary samples of the current PU.

Figure 3 presents the reference sample locations a and b relative to the current sample x to be predicted for horizontal and vertical angular prediction with negative and positive prediction angles. At most two reference samples are selected for each sample to be predicted in the current PU. Depending on the current sample location and the selected prediction angle, reference sample a and b can be neighboring PUs, padded samples, or samples inside the current PU. The interpolation for prediction sample generation is exactly same as that in lossy coding.



(a) Vertical sample-based angular prediction with positive angle

(b) Horizontal sample-based angular prediction with positive angle

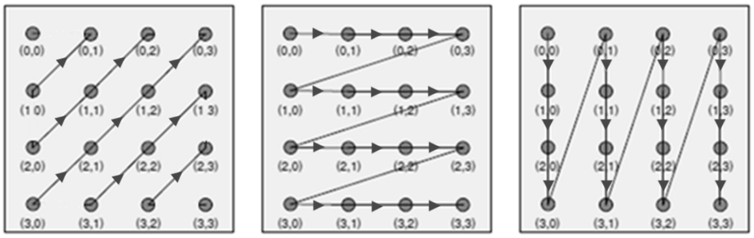
(c) Vertical sample-based angular prediction with negative angle

(d) Horizontal sample-based angular prediction with negative angle

Figure 3. Reference sample locations relative to the current sample for sample-based angular intra prediction

2.2. Mode dependent coefficient scanning

In HEVC intra coding, mode dependent coefficient scanning (MDCS) [14] is used. There are three scan patterns: diagonal [15], horizontal, and vertical, as shown in Figure 4. The each scanning pattern is represented by the scan index. Index 1 and index 2 are assigned for horizontal and vertical scans, respectively. For diagonal scan, index 3 is assigned. Scanning pattern for the current transform unit (TU) is determined by the intra prediction mode and the TU size using a fixed look-up table.



(a) Diagonal scan (b) Horizontal scan (c) Vertical scan

Figure 4. Three scanning patterns: diagonal, horizontal, vertical scans

Table 1 shows the look-up table that is used for the scan index selection. The look-up table is changed from the earlier version of MDCS. That is because the defined intra prediction mode number is changed in consecutive order. Here, the first row of the table indicates the intra prediction mode. The first column of the table represents the TU size. According to information of the intra prediction mode and the TU size, we can find the appropriate scan index using Table 1.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
32x32	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
16x16	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
8x8	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	1	1	1	1	1	1	1	1	1	3	3	3	3
4x4	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	1	1	1	1	1	1	1	1	1	3	3	3	3

Table 1. Look-up table for the intra coefficient scan index selection

2.3. Entropy coding

2.3.1. Syntax elements of CABAC

HEVC employed context-based adaptive binary arithmetic coding (CABAC) as an entropy coder. The syntax elements employed in CABAC are shown in Table 2. The gray shaded syntax elements are encoded in TU level and others are encoded in 4×4 sub-TU level.

last_significant_coeff_x_prefix
last_significant_coeff_y_prefix
last_significant_coeff_x_suffix
last_significant_coeff_y_suffix
significant_coeff_group_flag
significant_coeff_ flag
coeff_abs_level_greater1_flag
coeff_abs_level_greater2_flag
coeff_sign_flag
coeff_abs_level_remaining

Table 2. CABAC syntax elements for a transform unit (TU)

Last Significant Coefficient Position Coding: Since HEVC employs big coding unit up to 64x64, the location of the last significant coefficient in a TU is encoded by the column and the row position. For a TU larger than 4x4, the syntax element is separated into two parts: prefix and suffix. Prefix and suffix parts are encoded using truncated unary code and fixed length code, respectively. Table 3 shows the codeword structure for syntax elements of last significant coefficient position. In Table 3, (1) only exists when the TU size is greater than the largest last position that the code can represent and X means 0 or 1.

Magnitude of last coefficient position	Prefix (Truncated Unary Code)	Suffix (Fixed Length Code)
0	1	-
1	01	-
2	001	-
3	000(1)	-
4-5	00001	X
6-7	00000(1)	X
8-11	0000001	XX
12-15	0000000(1)	XX
16-23	000000001	XXX
24-31	000000000	XXX

Table 3. Codeword structure for syntax elements of last significant coefficient position

Significance Map Coding: After encoding of the position of last significant coefficient, significance map is encoded. There are two syntax elements, significant_coeff_group_flag and significant_coeff_flag. sgnificant_coeff_group_flag indicates that a 4x4 array of 16 transform coefficient level within the current TU has non-zero transform coefficient level. Then, for non-zero significant coefficient group, one bit symbol significant_coeff_flag is encoded in scanning order. If significant_coeff_flag is one, the transform coefficient level at the corresponding location has a non-zero value.

Level Information Coding: After the encoded significance map determines locations of all significant coefficients inside the TU, level information is encoded by using four syntax elements, including coeff_abs_level_greater1_flag, coeff_abs_level_greater2_flag, coeff_sign_flag, and coeff_abs_level_remaining. First two syntax elements indicate whether the quantized transform coefficient level value at the corresponding scanning position is greater than 1 and 2, respectively. Then, coeff_sign_flag is encoded. It specifies the sign of the coefficient. After this, the syntax element for the absolute value of the coefficient level minus three (coeff_abs_level_remaining) is binarized and encoded.

2.3.2. Binarization of level information

In order to binarize level information, the codeword is assigned as follows. Given a particular parameter *k*, an absolute transform coefficient *n* to be coded is consists of prefix part and a suffix part. The prefix is coded using a truncated unary code and the suffix is coded using a variable length code, as shown in Table 4. As shown in Table 4, the length of the variable length code depends on the unary code and the parameter *k*. That is, the parameter *k* controls the length of the codeword structure. Table 5 shows the binarization of coeff_abs_level_remaining when the parameter *k* is equal to 1.

Value	Prefix	Suffix (in bits)
$0 \sim 1\times2^{k}-1$	0	k
$1\times2^{k} \sim 2\times2^{k}-1$	10	k
$2\times2^{k} \sim 3\times2^{k}-1$	110	k
$3\times2^{k} \sim 4\times2^{k}-1$	1110	k
$4\times2^{k} \sim 5\times2^{k}-1$	11110	k
$5\times2^{k} \sim 6\times2^{k}-1$	111110	k
$6\times2^{k} \sim 7\times2^{k}-1$	1111110	k
$7\times2^{k} \sim 8\times2^{k}-1$	11111110	k
$8\times2^{k} \sim 9\times2^{k}-1$	111111110	k
$9\times2^{k} \sim 11\times2^{k}-1$	1111111110	$k+1$
$11\times2^{k} \sim 15\times2^{k}-1$	11111111110	$k+2$
$15\times2^{k} \sim 23\times2^{k}-1$	111111111110	$k+3$
…	…	…

Table 4. Binarization method for level information

Value	Prefix	Suffix
0	0	0
1	0	1
2	10	0
3	10	1
4	110	0
5	110	1
6	1110	0
7	1110	1
…	…	…
18	1111111110	00
19	1111111110	01
20	1111111110	10
…	…	…

Table 5. Example of binarization for level information when $k = 1$

The update of the parameter based on the magnitude of the previously encoded absolute level value. After encode one level value, the update mechanism is conducted, as shown in Eq. (1).

$$If\ \left|x\right| > 3 \cdot 2^{k},\ k' = \min(k+1,\ 4) \tag{1}$$

Here, x indicates the previously encoded level value, k is the parameter, and k' is the updated parameter. The parameter k ranged from 0 to 4. Based on the pseudo code, we can summarize the selected parameter according to the absolute level range.

Parameter	Absolute Level
0	0, 1, 2, 3
1	4, 5, 6
2	7, 8, ..., 12
3	13, 14, ..., 24
4	25, 26, ..., ∞

Table 6. Absolute level range for determining the parameter

In level information coding, the absolute value of each non-zero coefficient is adaptively encoded by a codeword structure with the selected parameter *k*. The codeword with certain parameter is designed to encode efficiently in a specified range of the absolute level, as described in Table 6. We can note that the parameter monotonically increases according to the previously encoded absolute level. That is because level coding in CABAC is based on the expectation that absolute level is likely to increase at low frequencies.

4. Efficient differential pixel value coding

In this section, we introduce an efficient differential pixel value coding method. The proposed method consists of two parts: mode dependent differential pixel scanning and level information coding with modified binarization.

4.1. Mode dependent differential pixel scanning

In the HEVC scanning method, the horizontal scan is used for a vertically predicted block. In the similar way, for a horizontally predicted block, the vertical scan is used. Undoubtedly, SAP significantly improves coding efficiency of intra prediction in lossless coding. However, since the current sample cannot exactly predicted by reference samples and there is no transform and quantization processes, correlation in the prediction direction still remains. Thus, the conventional scanning index mapping in HEVC cannot provide the best coding performance for lossless video coding.

In lossless coding, intra predicted residuals do not show the same behavior as transformed coefficients. Instead, it is observed that for relatively small TU, e.g. an 8x8 or a 4x4 TU, when intra prediction is in vertical direction, the residual will often appear in vertical direction. Thus, a vertical scan will often result in better performance. Similarly, when the intra prediction is in horizontal direction, a horizontal scan will often be better. It is motivation of MDRS [16] and we follow this observation.

We assign the vertical scanning pattern to the vertically predicted block and the horizontal pattern to the horizontally predicted block. However, MDRS is proposed for the HEVC test model (HM) 4.0 and the current HEVC standard uses the different intra prediction mode number. Hence, we change the scan index selection to fit the current HEVC intra prediction mode number, as shown in Table 7.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
32x32	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
16x16	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
8x8	3	3	3	3	3	3	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	3	3	3	3
4x4	3	3	3	3	3	3	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	3	3	3	3

Table 7. Modified look-up table for the scan index selection

In lossless coding, these differential pixel values are likely to be the end of the PU. As mentioned in Section 2, padded samples are produced and used as reference samples in the prediction process. Figure 5 shows an example that a padded sample is used as reference sample. Here, the padded samples are copied from the closest neighboring sample s. Strictly speaking, these padded samples are not actual neighboring samples of the current sample x and samples that uses these padded samples as reference samples might provide poor prediction performance. It results in the increase of the residual data.

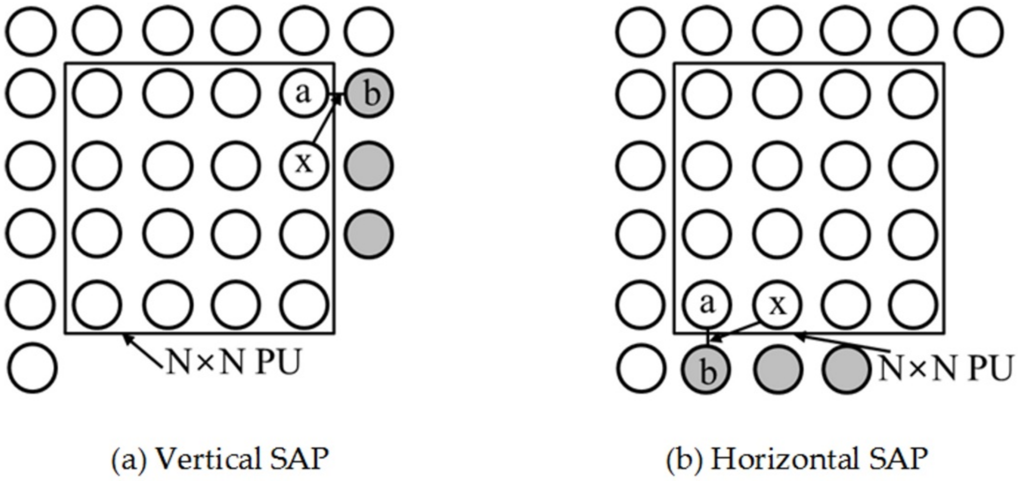


Figure 5. Two types of padded samples in the sample-based angular prediction

Since syntax elements in the entropy coder are encoded in the reverse order, the beginning part of the scanned coefficient sequence has a higher probability of having non-zero coefficients compared with the ending part. In this way, the resultant scanned sequence is more suitable for the entropy coding method and experimental results verify that considerable bit saving is achieved. Thus, we change the scan order. For each scanning pattern, we change the scan order in the opposite order of the conventional scanning method.

4.2. Level information coding with modified binarization

As mentioned, in lossless coding, the residual data is the differential pixel values between the original and the predicted pixel values without transform and quantization. Main difference between differential pixel values in lossless coding and quantization transform coefficients of lossy coding is the magnitude of the level information. Figure 6 shows the magnitude distribution of coeff_abs_level_remaining in lossy and lossless coding. We can observe that differential pixel values have much bigger level information than quantized transform coefficients in lossy coding. In other words, differential pixel values have a wide range of magnitudes.

Hence, in our binarization, we extend the parameter range from 0 to 6. The parameter is initially set to zero. The parameter monotonically increases based on Eq. (2).

$$If\ \ |x| > 3 \cdot 2^{k},\ k' = \min(k+1,\ 6) \tag{2}$$

It contributes the coding performance of the proposed method since it provides appropriate binarization code for bigger level values. The construction method of codeword is same with conventional HEVC coding, as described in Section 2.

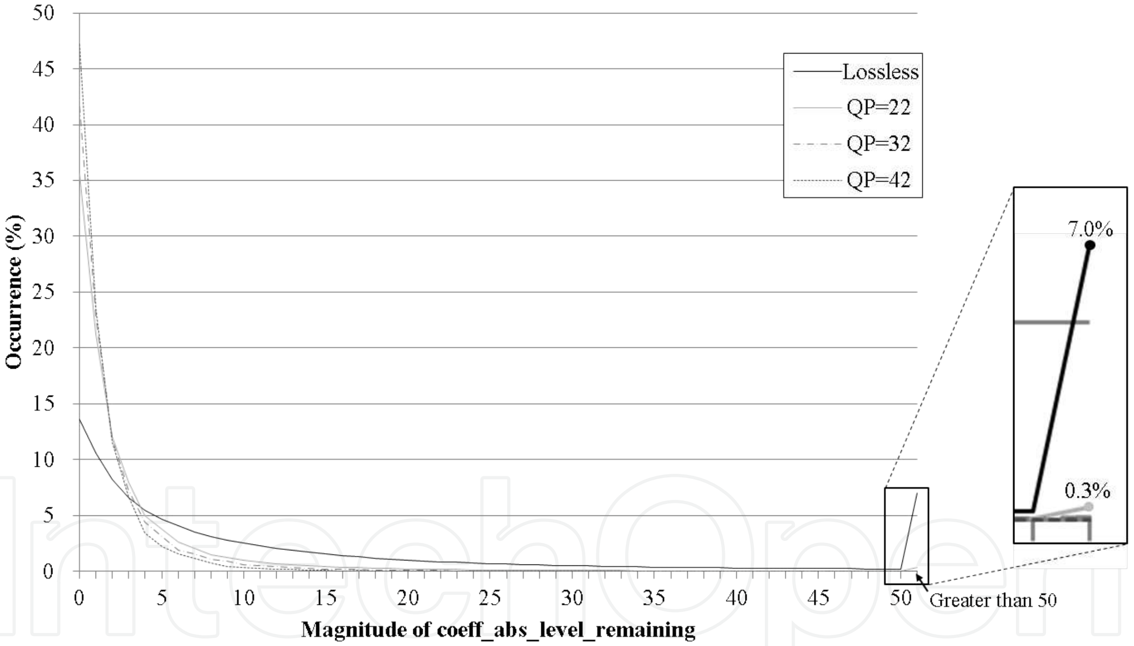


Figure 6. Magnitude distribution of coeff_abs_level_remaining

5. Experimental results and analysis

In order to verify coding efficiency of the proposed method, we performed experiments on several test sequences of YUV 4:2:0 and 8 bits per pixel format [17]. Two UHD (2560×1600) sequences, five HD (1920×1080) sequences, four WVGA(832×480) sequences, and four WQVGA(416×240) sequences with 100 frames are used. Specifically, the sequences that we used are summarized in Figure 7. The proposed method is implemented in HM 7.0 [18]. Table 8 shows the encoding parameters for the reference software.

Parameter	Value	Description
CUWidth	64	Largest CU size = 64x64
CUHeight	64	
IntraPeriod	1	Intra only coding
QP	0	Lossless coding
InternalBitDepth	8	8 bit per pixel
LosslessCuEnabled	1	Lossless coding
LoopFilterDisable	1	No deblocking filter
SAO	0	No sample adaptive offset
ALF	0	No adaptive loop filter

Table 8. Encoding parameters

In order to evaluate the efficiency of the proposed method, we include two sections based on the following settings.

- *MethodI*: Mode dependent differential pixel scanning
- *Method II*: *MethodI* + Entropy coding with modified binarization

5.1. Coding performance comparison

To verify the performance of the proposed method, we evaluate the compression results using bit saving. The definition of the measure is shown in Eq. (3). In the bit saving, negative value represents higher compression efficiency.

$$Bit\ Saving(\%) = \frac{Bitrate_{Method} - Bitrate_{HEVC-LS}}{Bitrate_{HEVC-LS}} \times 100 \tag{3}$$

(a) Traffic (UHD) (b) PeopleOnStreet (UHD) (c) Kimono (HD)

(d) ParkScene(HD) (e) Cactus (HD) (f) BasketballDrive (HD)

(g) BQTerrace(HD) (h) BasketballDrill (WVGA) (i) BQMall (WVGA)

(j) PartyScene (WVGA) (k) RaceHorses (WVGA) (l) BasketballPass (WQVGA)

(m) BQSquare (WQVGA) (n) BlowingBubbles (WQVGA) (o)RaceHorses (WQVGA)

Figure 7. HEVC common test sequences

Sequence	HEVC lossless mode (bytes)	Proposed Method *Method* I (bytes)	Proposed Method *Method* II (bytes)	Bit Saving of Method *I* (%)	Bit Saving of Method II (%)
Traffic	292043389	289021760	288978917	-1.03	-1.05
PeopleOnStreet	285283752	279453835	279453835	-2.04	-2.10
Kimono	135252302	135252302	134976664	-0.20	-0.21
ParkScene	162834309	162834309	161872833	-0.59	-0.60
Cactus	165415216	165415216	164741620	-0.41	-0.42
BasketballDrive	142954441	142954441	142590363	-0.25	-0.26
BQTerrace	159248773	159248773	158651984	-0.37	-0.43
BasketballDrill	29015828	29015828	28759282	-0.88	-0.90
BQMall	31047873	31047873	30892776	-0.50	-0.52
PartyScene	38443400	38443400	38237571	-0.54	-0.54
RaceHorses	31042618	31042618	30855443	-0.60	-0.59
BasketballPass	6938810	6938810	6853810	-1.22	-1.24
BQSquare	8940364	8940364	8893462	-0.52	-0.83
BlowingBubbles	9009542	9009542	8975937	-0.37	-0.37
RaceHorses	8309686	8309686	8249459	-0.72	-0.79
Average				**-0.69**	**-0.72**

Table 9. Comparison of bit savings for the HEVC lossless mode and the proposed method

Experimental results are presented in Table 9. It can be seen that the proposed method gives additional compression efficiency about 0.72% bit savings on average and 2.10% bit savings at maximum compared to the HEVC lossless mode. From Table 9, we confirmed that the proposed method provided better coding performance, compared to the conventional HEVC lossless mode.

5.2. Encoding time comparison

To verify the complexity of the proposed method, we check encoding time of the proposed method and the conventional HEVC lossless mode. Then, we calculate the encoding time change ($\Delta EncodingTime$), as defined in Eq. (4). Here, negative value means the complexity reduction and positive value means the complexity increase.

$$\Delta EncodingTime(\%) = \frac{EncodingTime_{Method} - EncodingTime_{HEVC-LS}}{EncodingTime_{HEVC-LS}} \times 100 \tag{4}$$

The complexity comparison results are presented in Table 10. In general, the most time consuming part in intra lossless coding is not the prediction part, but residual data coding. However, since the proposed method follows the statistical results of lossless coding and consists of simple operations, the variation of the complexity is typically small. It is shown that all encoding time increases are less than 0.65%. In some cases, the encoding time is rather decreased. The amount of decreased encoding time is 1.96% at maximum, compared to the HEVC lossless mode.

Sequence	**Proposed Method**	
	Method I	***Method II***
Traffic	+0.10	+0.05
PeopleOnStreet	-0.10	-0.78
Kimono	-0.31	-0.79
ParkScene	+0.65	-0.01
Cactus	-0.20	-0.01
BasketballDrive	-0.30	-0.06
BQTerrace	+0.51	-0.22
BasketballDrill	+0.33	-0.16
BQMall	+0.35	+0.18
PartyScene	-0.08	+0.23
RaceHorses	-0.05	-0.05
BasketballPass	-1.96	-1.16
BQSquare	-0.44	-0.39
BlowingBubbles	+0.18	-0.05
RaceHorses	-0.17	+0.50

Table 10. Encoding time change (%)

6. Conclusions

In this chapter, we proposed the improved differential pixel value coding method for HEVC lossless intra coding. Considering statistical differences in residual data between lossy and lossless coding, we designed new scanning and context-based adaptive binary arithmetic coding (CABAC) binarization methods. In the proposed scanning method, we used vertical scan for vertical prediction and horizontal scan for horizontal prediction. Besides, we changed the scan order in the reverse order. In the proposed binarization method, we extended the range of binarization parameter based on the observed statistical characteristics of residual data in lossless coding. Experimental results show that the proposed method provided approximately 0.72% bit savings without significant complexity increase, compared to HEVC lossless intra coding.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012-0009228).

Author details

Jung-Ah Choi* and Yo-Sung Ho

*Address all correspondence to: jachoi@gist.ac.kr

Gwangju Institute of Science and Technology (GIST), 261 Cheomdan-gwagiro, Buk-gu, Gwangju, Republic of Korea

References

[1] ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11. High efficiency video coding (HEVC) text specification draft 7, JCT-VC document, JCTVC-I1003, Geneva, CH, April 2012.

[2] Ho Y.-S., Choi J.-A. Advanced video coding techniques for smart phones," Proceedings of the International Conference on Embedded Systems and Intelligent Technology (ICESIT) 2012, 27-29 Jan. 2012, Nara, Japan.

[3] ISO/IEC JTC1/SC29/WG11. Vision, application, and requirements for high performance video coding (HVC), MPEG document, N11096, Kyoto, JP, Jan. 2010.

[4] Sayood K., editor. Lossless Compression Handbook. San Diego: Academic Press; 2003.

[5] ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11. AHG19: A lossless coding solution for HEVC, JCT-VC document, JCTVC-H0530, San José, CA, Feb. 2012.

[6] ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11. AHG19: A QP-based enabling method for lossless coding in HEVC, JCT-VC document, JCTVC-H0528, San José, CA, Feb. 2012.

[7] ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11. AHG22: Sample-based angular prediction (SAP) for HEVC lossless coding, JCT-VC document, JCTVC-G093, Geneva, CH, April 2012.

[8] ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11. AHG19: Method of frame-based lossless coding mode for HEVC, JCT-VC document, JCTVC-H0083, San José, CA, Feb. 2012.

[9] ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11. AHG22: A lossless coding solution for HEVC, JCT-VC document, JCTVC-G664, Geneva, CH, April 2012.

[10] ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11. AHG22: Lossless Transforms for Lossless Coding, JCT-VC document, JCTVC-G268, Geneva, CH, April 2012.

[11] ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11. Simplified CABAC for Lossless compression, JCT-VC document, JCTVC-H0499, San José, CA, Feb. 2012.

[12] ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11. JCT-VC AHG report: Lossless Coding (AHG13), JCT-VC document, JCTVC-I0013, Geneva, CH, April 2012.

[13] ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11. JCT-VC AHG report: Lossless Coding (AHG11), JCT-VC document, JCTVC-J0011, Stockholm, SE, July 2012.

[14] ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11. CE11: Mode Dependent Coefficient Scanning, JCT-VC document, JCTVC-D393, Daegu, KR, Jan. 2011.

[15] ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11. CE11: Parallelization of HHI_TRANSFORM_CODING (Fixed Diagonal Scan), JCT-VC document, JCTVCF129, Torino, IT, July 2011.

[16] ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11. AHG22: A lossless coding solution for HEVC, JCT-VC document, JCTVC-G664, Geneva, CH, April 2012.

[17] ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11. Common HM test conditions and software reference configurations, JCT-VC document, JCTVC-I1100, Geneva, CH, April 2012.

[18] JCTVC HEVC Test Model (HM). http://hevc.kw.bbc.co.uk/trac/browser/tags/HM-7.0