20-µs Low-latency Simple Video Coding with High Compression Ratio for IoT Video-transmission

Mai Yamakawa
Graduate School of Engineering and Technology
Nihon University
Fukushima, Japan
cema23021@g.nihon-u.ac.jp

Seiji Mochizuki
Faculty of Engineering
Osaka Sangyo University
Osaka, Japan
Fukushima, Japan
seiji.mochizuki@eic.osaka-sandai.ac.jp

Yu Kuwahara
Graduate School of Engineering and Technology
Nihon University
Fukushima, Japan
ceuu24006@g.nihon-u.ac.jp

Tsuyoshi Kato
Graduate School of Engineering and Technology
Nihon University
Fukushima, Japan
cetu21004@g.nihon-u.ac.jp

Kousuke Imamura
College of Science and Engineering
Kanazawa University
Ishikawa, Japan
imamura@ec.t.kanazawa-u.ac.jp

Tetsuya Matsumura
College of Engineering
Nihon University
Fukushima, Japan
matsumura.tetsuya@nihon-u.ac.jp

***Abstract* —. In recent years, low-latency video transmission has become necessary for IoT devices and autonomous driving systems. We previously reported a single-line-based ultra-low-latency video coding method to meet this requirement that achieves microsecond order latency for 4K/8K high-resolution video. This method achieved a compression ratio of less than 30% with a simple process. At the present time, however, for 5G-based video transmission in various current applications, a low-latency performance of 1 ms is practical, and a better compression ratio is consequently required. Therefore, a video coding method that satisfies the low latency of 10µs, which accounts for only a few percent of 1 ms video transmission, is urgently required, and a high compression ratio while maintaining high image quality, as well as low power consumption and low cost. In this paper, we present a video coding method that uses 4 lines as the basic processing unit and achieves a low latency of about 20 µs. This 4-lines-based processing includes three features to improve the compression ratio: a variable block size, adaptive selection of orthogonal transforms, and a new image prediction mode. The proposed method achieves an average compression ratio of 9.6% at 40 dB, a 2.2-percentage-point improvement over previous work. In addition, compared to the standard H.264 baseline profile, the proposed method achieves a computation scale of half or less and has advantages in terms of power consumption and implementation cost.**

Keywords—video coding, low latency, autonomous driving, virtual reality (VR), high-resolution video, 5G, H.264

I. Introduction

In recent years, low-latency video transmission between devices has become essential for applications requiring real-time motion control, such as IoT devices and autonomous driving systems. In general, transmission latency in the current 5G communication network is approximately 1 ms. In this environment, transmission latency of microseconds is essential for video transmission. However, conventional video coding standards such as H.264 [1] and H.265 [2] prioritize the realization of high compression rates at the expense of low latency to support narrow bandwidth and long-time recording. In contrast, standard H.264 defines a baseline profile that uses only I/P picture frames to achieve low latency. However, the processing of this profile is complex and requires a huge number of calculations, causing latency in the order of ms. In response to this challenge, an ultra-low-latency video coding method for Full-HD [3] was reported, in which the coding processing unit is not in rectangular units as in the standard, but in single-line units. This method processes 16 pixels × 1 line as the unit of coding, which enables millisecond-order latency. Furthermore, an ultra-low latency video coding method [4] was proposed for 4K/8K high-definition video, in which the processing unit is extended to 32 pixels × 1 line. The proposed method is a non-standard coding method with a simple structure, requires a small number of operations, and has been confirmed to improve the compression ratio. However, even with this line-based coding method, it is difficult to achieve a sufficient compression ratio. Achieving this ultra-low-latency performance on the order of microseconds is an essential issue for future applications. However, for video transmission applications in current 5G communications, a latency of several tens of microseconds, which does not adversely affect the 1-ms transmission latency of 5G, is sufficient for practical use. Figure 1 shows the target area of application fields. The compression ratio must be higher than the single-line-based coding, and the computational cost must be sufficiently lower than the H.264 baseline profile. The proposed coding method needs to achieve low latency that does not have an overly negative impact on the transmission latency of 1 ms in 5G communication networks. Therefore, this coding method covers the target area in Fig. 1. In other words, it must have a higher compression ratio than the single-line-based coding and a sufficiently lower computational cost than the H.264 baseline profile. In this study, we developed a new video coding method based on 4-line processing to meet the

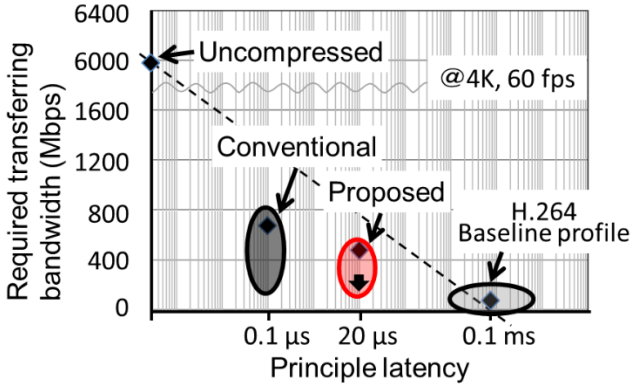


Fig. 1 Target area of application fields

performance of the target zone in Fig. 1, which achieves an ultra-low latency of about 20 µs. This method improves the compression ratio by extending the basic processing unit of coding to 32 pixels × 4 lines. Furthermore, three new techniques were incorporated: a variable block size for adaptively selecting the coding processing unit, adaptive selection of the orthogonal transformation method, and a new image prediction mode. The proposed method guarantees high compression and low-latency performance of 20 µs for 4K video coding. We compared our new method with the standard H.264 baseline profile and conventional methods in terms of compression ratio and the number of operations, which are highly dependent on hardware size and power consumption.

Section 2 presents an overview of the previous ultra-low-latency video coding method and the 4-line-based ultra-low latency video coding method, which extends the basic processing unit to 32 pixels × 4 lines, the basis of this paper. Section 3 describes the three new features added to the 4-line-based ultra-low-latency video coding method: variable block size, adaptive selection of the orthogonal transformation method, and new image prediction. In Section 4, the performance of the proposed method is evaluated in terms of image quality and compression ratio compared to conventional single-line-based ultra-low-latency video coding methods. Section 5 explains the degree of contributions of each of the three new features; variable block size, adaptive selection of orthogonal transformation schemes, and new image prediction. In Section 6, the computational costs of the proposed method and the H.264 baseline profile are comparatively evaluated by measuring the processing time required for each simulation. Finally, Section 7 concludes this paper in light of the evaluation results of Sections 4 and 6.

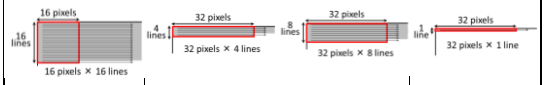
II. Basic Algorithm of the Proposed Method

The ultra-low latency video coding method [3, 4] employs single-line-based coding that reduces latency on the millisecond order to microsecond (i.e., to 1/1000) order compared to the conventional video coding method [1, 2]. Table I shows the relationship between coding processing units and latency in the proposed and conventional methods. H.264 uses 16 pixels × 16 lines as the basic processing unit for coding, and 15 lines + 16 pixels of pixel data are accumulated by the start of processing. As a result, the principle latency of 4K/60 fps video is 100 µs. In the ultra-low latency video coding method, coding is carried out in line units called compressed blocks (CBs), which require 32 pixels of data to be accumulated by the start of processing, resulting in a latency of 0.05 µs. In the proposed method, the CB is extended to 32 pixels × 4 lines, producing a latency of 20 µs until the start of processing. Thus, the proposed method achieves a practical ultra-low latency compared to conventional video coding methods.

III. Details of Proposed Method

Figure 2 shows a block diagram of the proposed ultra-low latency video coding method. Table II shows a comparison of the H.264 baseline profile and ultra-low latency video coding method specifications. The proposed method adopts three new techniques: an implementation of the variable block size method, adaptive selection of the orthogonal transform, and a new approach to image prediction (image prediction for variable block sizes).

TABLE I. Relationship between coding processing unit and latency

Item		H.264/AVC	This work		Previous work [4]
			4 lines	8 lines	
Coded processing unit					
4K	Stored data	3840 pixels × 15 lines + 16 pixels	3840 pixels × 3 lines + 32 pixels	3840 pixels × 7 lines + 32 pixels	32 pixels
	Latency	100 µs (60 fps) 200 µs (30 fps)	20 µs (60 fps) 40 µs (30 fps)	45 µs (60 fps) 90 µs (30 fps)	0.05 µs (60 fps) 0.10 µs (30 fps)
8K	Stored data	7680 pixels × 15 lines + 16 pixels	7680 pixels × 3 lines + 32 pixels	7680 pixels × 7 lines + 32 pixels	64 pixels
	Latency	50 µs (60 fps) 100 µs (30 fps)	10 µs (60 fps) 20µs (30 fps)	23 µs (60 fps) 46 µs (30 fps)	0.01 µs (60 fps) 0.02 µs (30 fps)

・Latency = Stored data (pixels) / Dot clock (MHz)
・Dot clock = 4K 60 fps: 594 MHz, 8K 60 fps: 2376 MHz,
30 fps: 297 MHz. 30 fps: 1188 MHz.

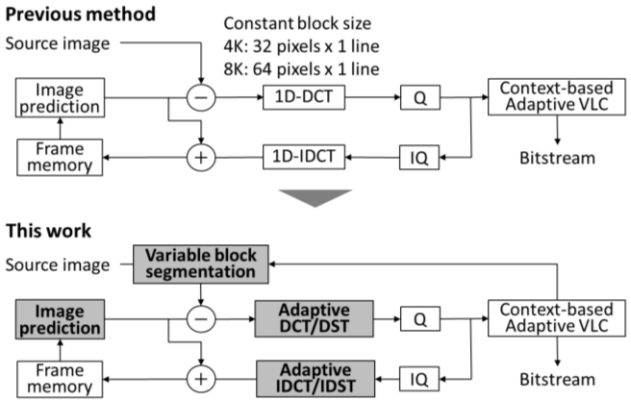


Fig. 2 Block diagram for proposed method

TABLE II. Comparison of H.264 baseline profile and ultra-low-latency video coding method specifications

Item		H.264/AVC baseline profile	This work		Previous work [4]
			4 lines	8 lines	
CB (variable block size)	4K	16×16,16×8, 8×16,8×8, 8×4,4×8,4×4	8×4, 16×4, 32×4, 64×4	8×8, 16×8, 32×8, 64×8	32×1
	8K		16×4 ,32×4, 64×4,128×4	16×8, 32×8, 64×8,128×8	64×1
Prediction	Intra	9 modes	7 modes		4 modes
	Inter	Reference	Same position as in the previous frame		
Transform	4K	2D-DCT 4×4, 8×8	2D-DCT/DST 4×4	2D-DCT/DST 8×8	1D-DCT 32×1
	8K				1D-DCT 64×1
Quantization		H.264-based quantization			
Entropy coding		Context-adaptive variable-length coding (CAVLC)			Line-based CAVLC
Others		Deblocking filter/error resilience			

A. Variable block size

The variable block size method reduces the amount of information by adaptively adjusting CBs based on the spatial redundancy of the video image. Figure 3 shows the algorithm for variable block size determination. In our method, the CB is determined using a binary tree partitioning algorithm. Four different CBs are selected for the coding of a 4K video image with four lines: 64 pixels × 4 lines, 32 pixels × 4 lines, 16

pixels × 4lines, and 8 pixels × 4 lines. The evaluation function for determining the CB is the sum of absolute difference (SAD) after image prediction. In Figure 3, the SAD of one 64-pixel × 4-line CB and two 32-pixel × 4-line CBs are compared at Depth 0, and if the SAD of one 64-pixel × 4-line CB is small, the coding process is performed with 64 pixels × 4 lines without division. However, if the SAD of two 32-pixel × 4-line CBs is small, the division is performed and the process moves to the Depth 1 layer to make the division decision again. This makes it possible to select large CBs such as 32 pixels × 4 lines or 64 pixels × 4 lines in areas with high spatial redundancy, and small CBs such as 8 pixels × 4 lines or 16 pixels × 4 lines in areas with low spatial redundancy. As a result, high compression coding is achieved according to the characteristics of each picture. Adopting the new variable block size method makes it necessary to identify the type of CB when decoding at the decoder. Therefore, two additional bits are added to specify four different CBs.

B. Adaptive selection of DCT/DST

The H.265 standard employs two types of orthogonal transforms [5], the discrete cosine transform (DCT) and discrete sine transform (DST), with the DST being used fixedly only for the coding process of 4-pixel × 4-line luminance signals. Our method employs an adaptive orthogonal transform selection method that selects between DCT-2 and DST -7 orthogonal transforms for each CB. The DCT and DST are shown in Equations (1) and (2), respectively. DST has characteristics opposite to those of DCT, and when the correlation with neighboring pixels is low, DST is more efficient than DCT for coding. Therefore, adaptive selection of orthogonal transforms is performed only when intra-prediction of luminance signals is used. Figure 4 shows the block sizes for orthogonal transforms. In this method, a 4-pixel × 4-line DCT and DST are used for 4-line coding, and the sum of squared errors (SSE) of the transformation coefficients is used as the evaluation formula for selecting the DCT and DST. As a selection method when CB is 32 pixels × 4 lines, the SSE when DCT processing is performed on all eight blocks is compared with the SSE when DST processing is performed on all the same eight blocks. The orthogonal transformation with the smaller SSE value is then adopted as the orthogonal transformation method of the CB. Adopting the adaptive selection of DCT/DST makes it necessary to identify the type of orthogonal transform when decoding at the decoder. Therefore, one additional bit is added to specify two different orthogonal transforms.

$$F(u) = \sqrt{\frac{2}{N}} C(u) \sum_{x=0}^{N-1} f(x) \cos\left(\frac{u\pi(2x+1)}{2N}\right) \quad (1)$$

$$u = 0,1,\ldots,N-1\text{c}$$

$$C_{(u)} = \begin{cases} 1 & u \neq 0 \\ \sqrt{\frac{1}{N}} & u = 0 \end{cases}$$

$$F(u) = \sqrt{\frac{4}{2N+1}} \sum_{x=0}^{N-1} f(x) \sin\left(\frac{\pi(2x+1)(u+1)}{2N+1}\right) \quad (2)$$

$$u = 0,1,\ldots,N-1\text{c}$$

C. Image prediction for variable block size

In our method, a total of eight prediction modes, consisting of seven effective intra-image prediction modes and one inter-image prediction mode, were selected to accommodate the extended CB size. Figure 5 illustrates the seven intra-image prediction methods using 16 pixels × 4 lines as an example. All prediction modes correspond to all four CBs required for the variable block size method. For the inter-

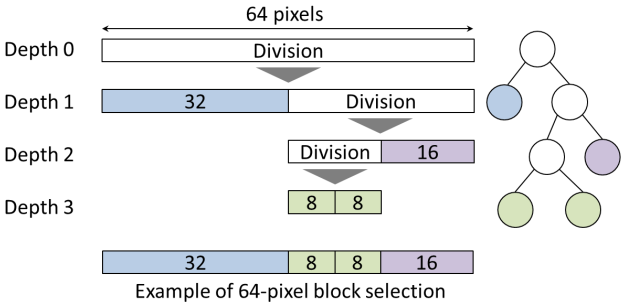


Fig. 3 Algorithm for determining variable block size

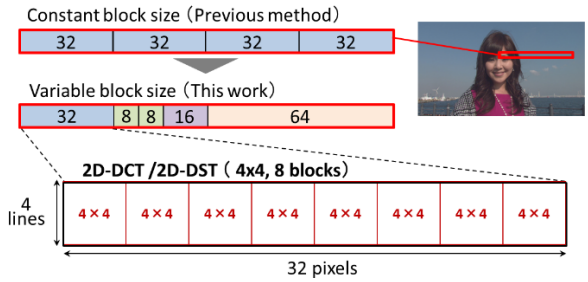


Fig. 4 Block size for orthogonal transforms

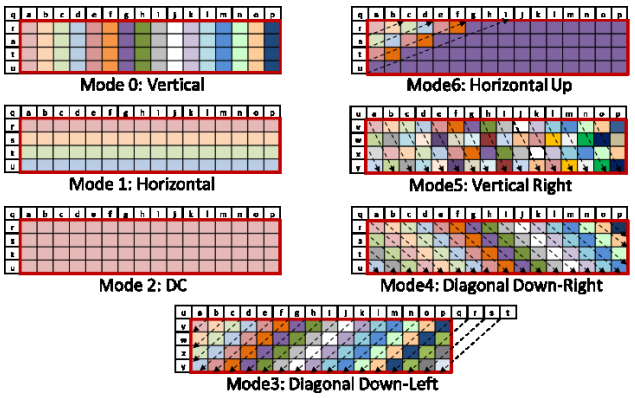


Fig. 5 Intra-prediction modes

image prediction mode, the same positional predictions that contribute most to coding efficiency were adopted.

IV. Evaluation and Experimental Results

Figure 6 shows the six standard 4K resolution video (3840 pixels × 2160 lines) sequences used in the evaluation. Steel Plant, Festival, Maiko, Layered Kimono, Marathon, and Horse Race. These are exhaustive sequences in the evaluation of encoding methods, as they have a variety of pictures, including festival scenes and people. The color space of the video sequence used for the evaluation is YUV420 with a depth of 8 bits. A frame rate of 60 fps is assumed. The image quality peak signal-to-noise ratio (PSNR) in decibels and percent compression ratio (CR) ratings are the average of these six standard video sequences. The simulations were performed with two frames, and the data were obtained with five quantization parameter steps of 15, 20, 25, 30, and 35. Figure 7 shows the average CR and PSNR values for each standard video sequence, expressed as a rate-distortion curve, and compares the results of the proposed and previous methods. Figure 7 shows that the previous method achieves a

Fig.6 4K resolution video sequences used in the evaluation

CR of about 11.8% at 40 dB, while our new method achieves a CR of about 9.6% at 40 dB.As a result, a CR improvement of about 2.2 percentage points (pt) is achieved. The sequence that showed the most CR improvement compared to the previous method was Festival. This sequence showed a CR improvement of about 2.6 pt at 40 dB.

V. Contribution of the Three Methods

In this section, we present the degree of contribution of each of the three newly proposed methods to the compression ratio as shown in Table Ⅲ. The listed values are the number of compression ratio improvement points for each method at 40 dB. In this evaluation, the most effective method of the three new proposals is the adaptive selection of orthogonal transforms. The next effective method is the new image prediction method, and the least effective method is the variable block size method.

VI. Discussion

In general, CR and PSNR values in coding have a strong correlation with the number of operations. Our new encoding method needs to be compared with the conventional H.264 baseline profile to determine whether fewer operations can achieve highly efficient compression. In this study, we measured the processing time in the simulation as an indicator of the amount of computation for each method. Table IV shows the average processing time of the H.264 baseline profile and the proposed method. As a result, the value of 4.5 was obtained for the proposed method when the processing time of the H.264 baseline profile was set to 10, which is less than half of that of the H.264 baseline profile. In addition, the number of operations was approximately 3.5 times greater than that of the conventional method. The new simulator is an experimental version and has not been optimized, while the H.264 reference software used in this study has been optimized through numerous upgrades. Therefore, it can be said that the actual number of calculations will be fewer than the value estimated from the actual measured processing time. This means that the proposed method can achieve ultra-low latency of 20μs compared to the H.264 baseline profile, and achieve higher compression than the conventional method

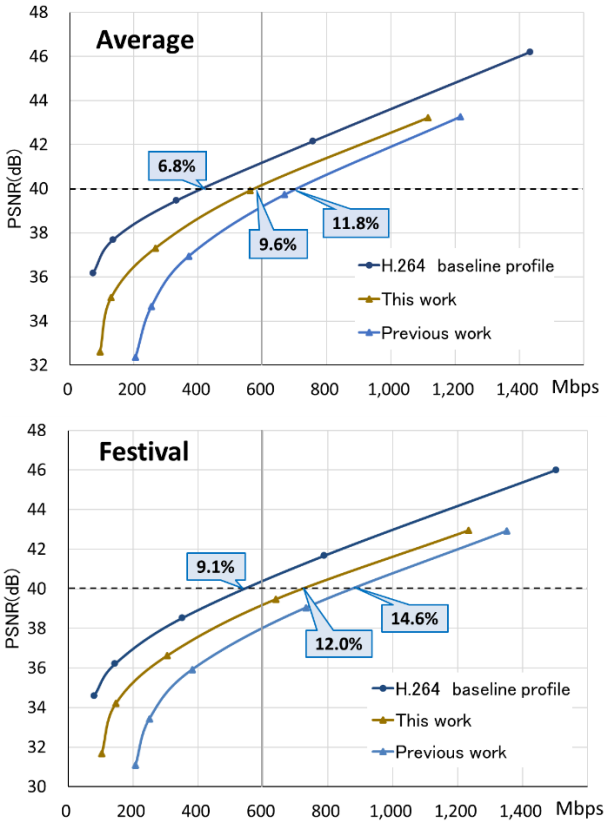


Fig. 7 Rate-distortion curves of the compared methods

TABLE III. The degree of contribution of each of the three newly proposed methods [pt]

Video sequence	Variable block size	Adaptive selection of DCT/DST	New image prediction
Steel Plant	2.26	2.42	2.40
Festival	1.94	2.48	2.33
Maiko	1.78	2.01	1.96
Layered Kimono	1.75	2.17	1.92
Marathon	1.96	2.75	1.43
Horse Race	1.15	1.23	1.23
Average	1.8	2.2	1.9

TABLE IV. Processing time for H.264 and the proposed method [s]

Video sequence	H.264 baseline profile	4-line based (this work)		1-line based (previous work)	
Steel Plant	118.27	51.23	4.3	13.94	1.2
Festival	113.27	51.24	4.5	13.94	1.2
Maiko	110.19	49.98	4.2	13.93	1.3
Layered Kimono	109.15	50.89	4.5	13.91	1.3
Marathon	109.78	50.84	4.6	13.90	1.3
Horse Race	109.78	51.73	4.7	13.90	1.3
Average*	**10**	**4.5**		**1.3**	

* Normalized H.264 baseline profile processing time as 10
(The simulation platform; windows-based PC consisting of a Core i9 (3.0 GHz) and 128 GB memory)

while achieving a unique non-standard codec with approximately half as much hardware or less.

VII. Conclusions

In this study, we developed an ultra-low-latency video coding method that achieves high image quality and a high compression ratio with an ultra-low latency of 20 μs. The proposed method extends the processing unit to 32 pixels × 4 lines and adopts three new techniques: a variable block size method, adaptive selection of orthogonal transform DCT/DST,

and a new intra-image prediction method. These techniques achieve a compression ratio improved by approximately 2.2 pt on average and up to 2.6 pt over conventional methods in coding 4K video images. In addition, a comparison of the processing time with the H.264 baseline profile showed that the computational cost is less than half that of the H.264 baseline profile, indicating that our new method has an advantage in hardware implementation. Therefore, this method is extremely effective for high-performance video coding in applications where low latency is required.

REFERENCES

[1] ITU-T Rec. H.264 (05/2003) Advanced video coding for generic audiovisual services, May. 2003, https://www.itu.int/rec/T-REC-H.264.

[2] ITU-T Rec. H.265 (08/2021) High-efficiency video coding, Aug. 2021, https://www.itu.int/rec/T-REC-H.265.

[3] S. Mochizuki, et al, "Ultra-low-latency Video Coding Method for Autonomous Vehicles and Virtual Reality Devices," IEEE IoTaIS 2018 pp.155-161, Feb. 2016.

[4] M. Yamaguchi, et al, "Ultra-low-latency Video Coding with Reduced Frame Memory Structure for 4K/8K High-Resolution Video," Proceedings of IEEE GCCE 2023, pp. 852-853. 2023

[5] A. Saxena, et al. "DCT/DST-Based Transform Coding for Intra Prediction in Image/Video Coding," IEEE Transaction on Image Processing, Vol. 22, No. 10, pp.3974-3981, Oct. 2013