

Regular Paper

Efficient Coding for Video Including Text Using Image Generation

YOSUKE NOZUE^{1,†1,a)} TOMO MIYAZAKI^{1,b)} YOSHIHIRO SUGAYA^{1,c)}
SHINICHIRO OMACHI^{1,d)}

Received: June 27, 2015, Accepted: December 7, 2015

Abstract: Text in video compressed by lossy compression at a low bitrate will easily be deteriorated, resulting in blurred text and a lower readability. In this paper, we propose a novel image coding method to preserve the readability of text in the video at a very low bitrate. During the encoding process, we estimate the parameters for each character of the text. Then, an image without text is generated and compressed. During the decoding process, we reconstruct video sequences with text from images without text and character images generated by the estimated parameters. The experimental results show the effectiveness of the proposed method in terms of the readability at a very low bitrate.

Keywords: image coding, character recognition, image generation

1. Introduction

Scene text provides various types of important information. There are many demands that transmit or distribute videos including text. However, the quality of the text in digital videos will deteriorate when it is highly compressed since it consists of high-frequency components. Nowadays, mobile video traffic exceeds 50% of the total amount of mobile data traffic, and it is expected that three-fourths of mobile data traffic will be video traffic by 2019 [1]. In addition, it is also important to be able to transmit important information such as text by video in a heavily congested situation such as during a disaster. Moreover, there is a demand to save as many videos as possible on devices with a small capacity such as an SD card. From these viewpoints, the realization of low-bitrate video encoding while maintaining the quality of important information such as text is an important research topic. We focus on coding for scene images including text in this study.

H.264 [2] is widely used for many applications including the broadcasting of high-definition TV. High Efficiency Video Coding (HEVC) [3] was proposed as the latest video coding standard, and it requires approximately 50% of the bitrate of H.264 to achieve an equivalent perceptual quality. Although these techniques achieve high performance for natural images, artificial content such as text in natural scenes will be degraded and result in a lower readability at a very low bitrate.

One technology for compressing video including text is screen content coding. Screen content is necessary for recent net-

worked applications such as cloud computing, shared screens, virtual desktops, etc. A typical computer screen image contains continuous-tone content such as text, icons, and graphics. Hu et al. proposed a method for preserving edges by introducing edge mode prediction besides intra prediction with the correlation of adjacent pixels [4]. Wang et al. proposed efficient block-based coding using the fact that similar blocks appear multiple times in an image [5]. Yang et al. applied a similar idea to the intra frames of videos [6]. The method proposed by Ma et al. used a color table for blocks including text for generating a predicted image [7], which is effective when the number of colors is limited. However, the effectiveness is limited when an object exists in two or more blocks because these techniques are applied to each block. Moreover, natural images including artificial objects such as text and buildings have more colors than the screen content. Therefore, screen content coding is not always effective for natural images including text consisting of many colors.

Another coding strategy related to the proposed method is object-based coding. The MPEG-4 standardized in 1998 enabled object-based video coding [8]. Anh et al. proposed a method for image coding by reducing the size of unnecessary objects by seam curving [9]. Deng et al. proposed a method for deleting the area including objects unnecessary for humans by introducing the block-based seam energy [10]. Ndjiki-Nya et al. changed the encoding methods for background regions by utilizing rigid and nonrigid textures [11]. Dumitras and Haskell drastically reduced the size while preserving textures by dividing the background according to the textures [12]. The method proposed by Azis et al. used the technique of restoring areas other than the region of interest (ROI) using the time correlation between frames [13]. Zhang and Bull proposed a method for selecting an appropriate encoding method by dividing images into foreground and background areas and classifying the background into dynamic and static parts [14]. A technique for reducing the amount of infor-

¹ Graduate School of Engineering, Tohoku University, Sendai, Miyagi 980-8579, Japan

^{†1} Presently with Panasonic Corporation

^{a)} nozue@iic.ecei.tohoku.ac.jp

^{b)} tomo@iic.ecei.tohoku.ac.jp

^{c)} sugaya@ecei.tohoku.ac.jp

^{d)} machi@ecei.tohoku.ac.jp

mation by using a graph [15] will be useful for dealing with arbitrarily shaped objects.

In this paper, we propose a novel method for coding images including text more efficiently while preserving the readability of the text by introducing the idea of object-based coding. During the encoding process, text areas are removed from the image and are filled with the background values. The video sequence without text is encoded. Meanwhile, a parameter that represents each character is estimated from the text areas. During the decoding process, the text areas are reconstructed with the character parameters, and they are combined with the background image. As a result, we can obtain a video including the restored text. The effectiveness of the proposed method is experimentally verified. The readability of text is verified by a subjective evaluation with the mean opinion score. The coding efficiency is compared with existing methods with the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM).

2. Proposed Method

2.1 Outline

The outline of the proposed encoder and decoder is shown in Fig. 1. The encoder detects text from video sequences with OCR and estimates the parameters of the detected text. Then, the text area is replaced by the background color. This image is called the difference image. Finally, HEVC encoding of the difference image is performed, and it is output with the parameters of the text.

During the decoding process, video data and the parameters of the text are input. The video sequences are reconstructed from these data.

2.2 Text Detection

The first step of the proposed method is text detection from the video sequences. After detecting text, the characters in the text are recognized, and the attributes of each character are determined. The attributes will be used during the decoding process. However, the methodology of text detection or character recognition itself is beyond the scope of this study since the purpose

of this research is to develop a coding method. There are many studies on detecting text from a scene image [16], [17], and we can also use commercial OCR software. In the experiment described in Section 3, we used commercial OCR software for text detection and recognition.

The text areas are tracked by a template matching technique using the sum of the absolute difference (SAD) defined by

$$SAD = \sum_x \sum_y |I_1(x, y) - I_2(x, y)|, \quad (1)$$

where $I_1(x, y)$ and $I_2(x, y)$ are intensity values at (x, y) of the two images.

2.3 Parameters of the Character Image

The parameters of the character image are estimated from each character in the text area. Let the parameters of character P be

$$P = \{x_p, y_p, c, s, \delta, f, b, \beta\}. \quad (2)$$

The meaning of each element is summarized in Table 1. In the third column of the table, the range of each value used in the experiments is given. Note that we only deal with a simple transformation of characters and images with a monochromatic background at the present stage. In general, we should also consider the affine or projective transformation and a more complex background, which requires more parameters. The consideration of such a general case is planned for future work.

The location of a character in the entire image is defined by (x_p, y_p) measured in pixels. The category c is the result of the character recognition described in Section 2.2. We use 26 alphabetical capital letters, 26 alphabetical small letters, and 10 digits. The size s is defined by the vertical size of a character. For the character font δ , we use 256 types of fonts. The foreground color f and background color b are determined by the RGB values. The blurred image is generated with a Gaussian filter. The degree of blur β is the standard deviation. Character images with various values of these parameters are shown in Fig. 2.

The parameters of each character are estimated by calculating the correlation of the input image and the images generated by changing a parameter. The parameter that has the highest correlation is selected. The algorithm is as follows.

- (1) First, the location, size, and font of each character are estimated. Given the detected character area, the search area is determined so that the width and height are twice the detected area, as shown in Fig. 3. By changing the location, size, and font, a character image is generated and the value of the SAD is calculated. The parameter that gives the minimum SAD is selected.
- (2) The foreground and background colors are estimated by

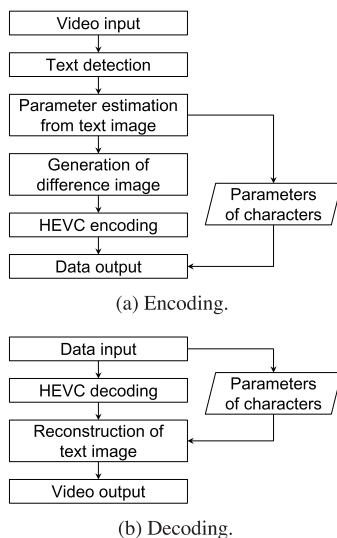


Fig. 1 Outline of the encoder and decoder.

Table 1 Parameters of a character.

parameter	meaning	range
x_p	horizontal location in the image	0 – 1279
y_p	vertical location in the image	0 – 719
c	category	0 – 61
s	size	0 – 127
δ	font	0 – 255
f	color of the character	0 – 255
b	color of the background	0 – 255
β	degree of blur	0 – 15



Fig. 2 Character images for various parameter values.



Fig. 3 Search area around the detected character area.

clustering pixels into two clusters by the K -means method.

- (3) Given the location, size, font, and foreground and background colors, the degree of blur is estimated for each character. By changing the value of β , character images with various degrees of blur are generated. The parameter β that gives the minimum SAD is selected as the parameter.

2.4 Generation of a Difference Image

A difference image is generated from the original image by

$$O'(x, y) = \begin{cases} \mathbf{b} + \mathbf{f} - \mathbf{O}(x, y) & \text{if } M(x, y) = 1 \\ \mathbf{O}(x, y) & \text{otherwise} \end{cases}, \quad (3)$$

where $\mathbf{O}(x, y)$, $\mathbf{O}'(x, y)$, \mathbf{f} , and \mathbf{b} are three-dimensional vectors that have RGB values in the range of 0 to 255. When a value is less than 0 or greater than 255, it is regarded as 0 or 255, respectively. $\mathbf{O}(x, y)$ represents the pixel at (x, y) of the input image, whereas $\mathbf{O}'(x, y)$ is the difference image. The values of \mathbf{f} and

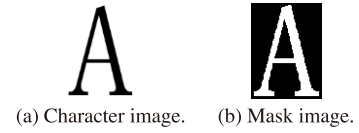


Fig. 4 Mask image.

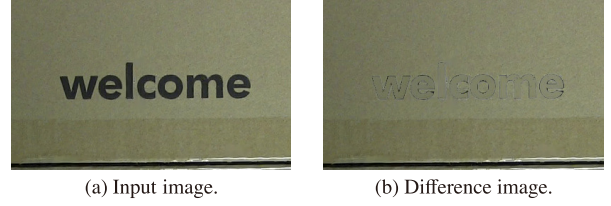


Fig. 5 Difference image.

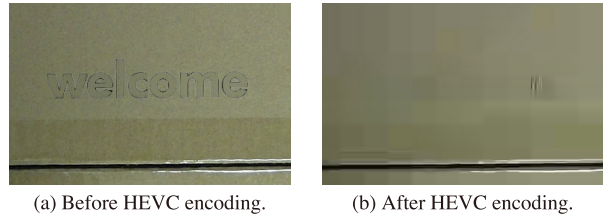


Fig. 6 HEVC encoding.

\mathbf{b} represent the foreground and background colors, respectively. $M(x, y)$ is the mask image that represents the foreground pixels of a character image defined by

$$M(x, y) = \begin{cases} 1 & \text{if } G^P(x, y) = \mathbf{f} \\ 0 & \text{otherwise} \end{cases}, \quad (4)$$

where $G^P(x, y)$ represents the image generated with the estimated parameter. An example of mask image is shown in Fig. 4. When $M(x, y) = 1$ is satisfied, the foreground object is removed by the parameter \mathbf{f} , and the parameter \mathbf{b} covers the removed area. As a consequence, the foreground object disappears in the difference image, and a uniform background remains. This process aids in efficient encoding.

The input and difference images are shown in Fig. 5. By this procedure, the edges of the characters can be removed, and the bitrate can be reduced.

2.5 Encoding

During the encoding process, we deal with the difference image and text information separately. HEVC encoding is applied to the generated difference image. Figure 6 displays an example.

For the text information, we create a file containing the parameters of every single character for each frame. Then, we compress the files with 7-zip compression software^{*1}. The overall bitstream that is transmitted to the decoder consists of encoded difference images and compressed text files.

2.6 Decoding

Given the encoded difference image and parameter values, an image is reconstructed. First, the difference image is decoded by the ordinary HEVC decoder. Then, the character images are reconstructed by the decoded difference image and parameter values using

^{*1} <http://www.7-zip.org/>

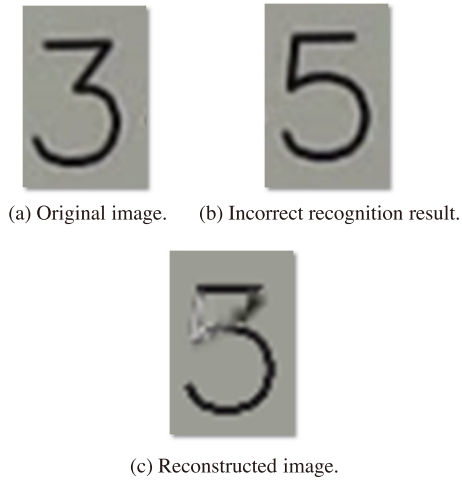


Fig. 7 Reconstruction of an image.

$$\hat{O}(x, y) = \begin{cases} \mathbf{b} + \mathbf{f} - \mathbf{O}''(x, y) & \text{if } M(x, y) = 1 \\ \mathbf{O}''(x, y) & \text{otherwise} \end{cases}, \quad (5)$$

where $\mathbf{O}''(x, y)$ represents the decoded difference image, and $\hat{O}(x, y)$ is the reconstructed image. This procedure enables the reconstruction of an image that is similar to the original one, even when the image \mathbf{G}^P is not ideally generated or the result of character recognition is incorrect. For example, in the case in which character recognition fails, the proposed method generates a difference image by changing the background pixels, as defined in Eq. (3), while the foreground pixels remain. Hence, the difference image becomes an undesired one. However, if we substitute $\mathbf{O}' = \mathbf{b} + \mathbf{f} - \mathbf{O}(x, y)$ in Eq. (3) for \mathbf{O}'' in Eq. (5), $\hat{O} = O$ will be obtained regardless of the mask values. Consequently, the input image can be reconstructed. An example of reconstruction is displayed in Fig. 7.

3. Experimental Validation

3.1 Experimental Setup

Experiments were carried out to verify the effects of the proposed method with regard to the following two aspects.

- Objective evaluation of the compression effect
- Subjective evaluation of the readability of the compressed video

For the objective evaluation, a rate–distortion curve (R–D curve) was used. In the R–D curve, the horizontal axis represents the bitrate, and the vertical axis represents the PSNR. The method with an R–D curve closer to the upper-left corner is better. The SSIM [18] was also used as another criterion. The SSIM is said to give a more subjective evaluation than the PSNR. For the qualitative evaluation, the mean opinion score (MOS) was used. The readability of the text in the generated video was subjectively evaluated by giving a rating from 1 (bad) to 5 (excellent). The ratings from twelve subjects were averaged to calculate the MOS. To verify the statistical significance of the difference between the MOSs for the proposed method and the existing method, Student's *t*-test with significance level of 0.05 was used.

A PC with a Core i5-2410M CPU running at 2.30 GHz with 4 GB of memory was used. Further, a Canon iVIS HF M52 video camera was used. The size of each frame was 1280×720. The six

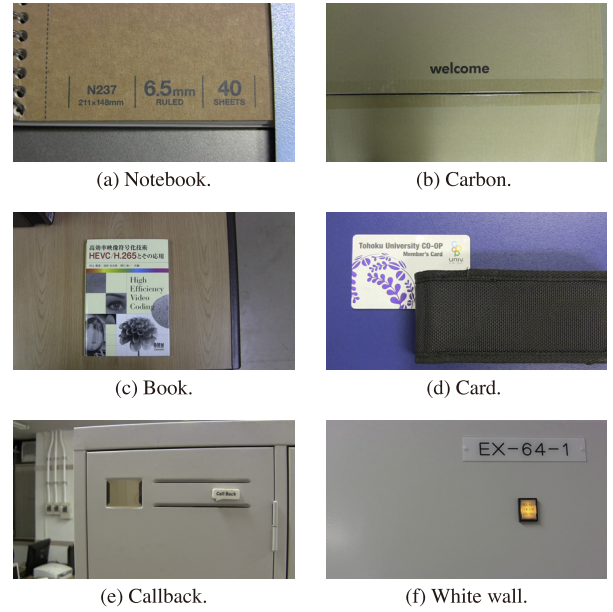


Fig. 8 Experimental data.

Table 2 Number of characters and the average size of the compressed parameters for a frame.

Data	Number of characters	Average size
(a)	29	279.5 bytes
(b)	7	65.4 bytes
(c)	33	291.8 bytes
(d)	31	273.0 bytes
(e)	8	94.5 bytes
(f)	5	50.1 bytes

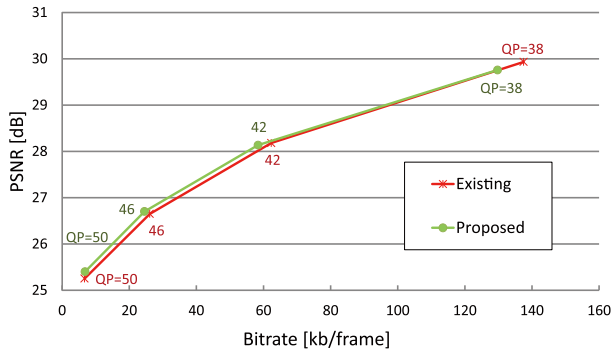
video sequences displayed in Fig. 8 were taken and used for evaluation. For each video, all of the frames include the same text. The text area was detected by commercial OCR software, and the recognition result was manually corrected when it was incorrect to purely evaluate the encoding performance. These conditions are reasonable because in general, encoding can be performed manually. The number of characters and the average size of the compressed parameters for text information for a frame are summarized in Table 2.

For comparison, the standard HEVC software HM-12.1 [19] was used as the existing method. The settings of the software were as follows.

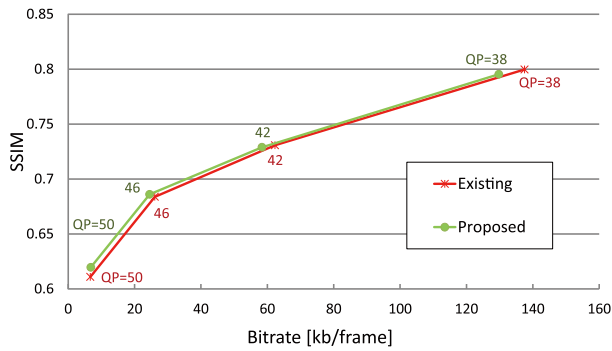
- Use only I-frames.
- QP values: 50, 46, 42, and 38.

3.2 Results and Discussion

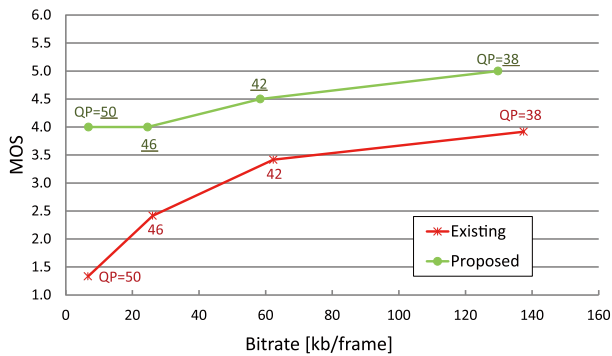
Figure 8 (a) shows a video sequence of a notebook. The PSNR, SSIM, and MOS are shown in Fig. 9 (a), (b), and (c), respectively. In the graph of the readability, the QP value is underlined if the difference in the MOS values between the proposed method and the existing method for the same QP are statistically significant. The PSNR and SSIM of the proposed method are almost the same as the existing method (standard HEVC encoding). However, the readability of the video generated by the proposed method is much higher than that of the existing method regardless of the QP values. In this case, there are statistically significant differences between the MOSs for all QP values. Expanded images of the text area in this video are shown in Fig. 10. At a very low



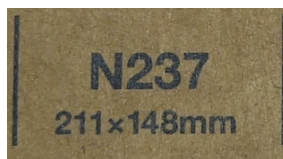
(a) PSNR.



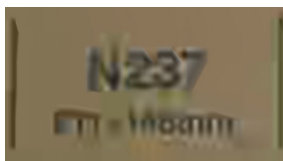
(b) SSIM.



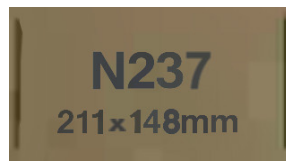
(c) Readability.

Fig. 9 Result 1 (Notebook).

(a) Original image.



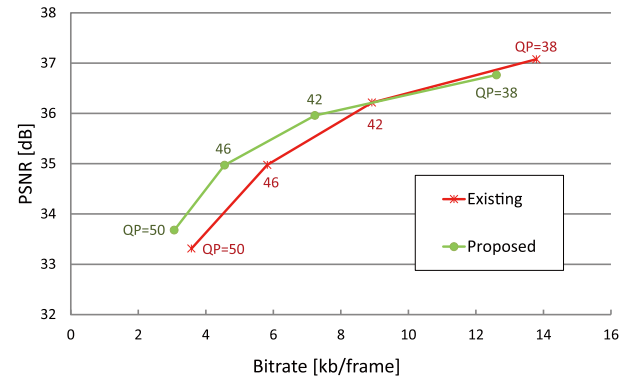
(b) Existing method (QP= 50).



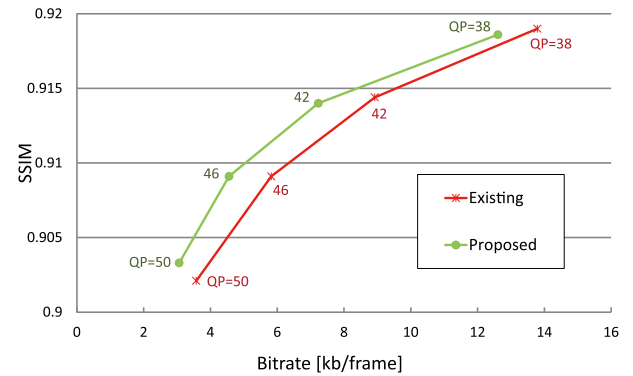
(c) Proposed method (QP= 50).

Fig. 10 Expanded image (Notebook).

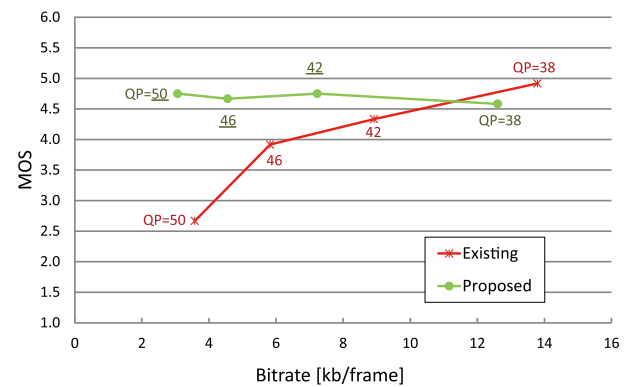
bitrate (QP = 50), the image for the existing method was heavily blurred, as shown in Fig. 10 (b), whereas the text area was finely reconstructed, as shown in Fig. 10 (c). At a relatively high bitrate, the MOS for the proposed method was also higher than that of the existing method. The reason is that the readability of the text



(a) PSNR.



(b) SSIM.

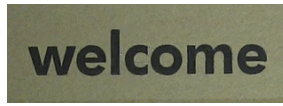


(c) Readability.

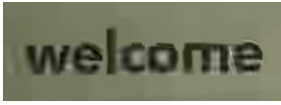
Fig. 11 Result 2 (Carbon).

in the original image was not very high since the image had some reflection caused by a fluorescent light, and the background of the notebook was fibrous, which made the character boundaries unclear. In the proposed method, the background was smoothed by compression, and the character boundaries become clearer than the original image.

The results for Fig. 8 (b) are shown in **Fig. 11**. At a low bitrate, the readability of the proposed method was higher than that of the existing method. In addition, the PSNR and SSIM were also higher than those of the existing method. Even in the case of QP = 38, where the MOS of the existing method was higher, there was no statistically significant difference. Although the PSNR for the proposed method was slightly worse than that of the existing method, the SSIM for the proposed method was better than that of the existing method. In this case, the text was finely reconstructed by the proposed method. On the other hand, the image was heavily deteriorated by the existing method, as



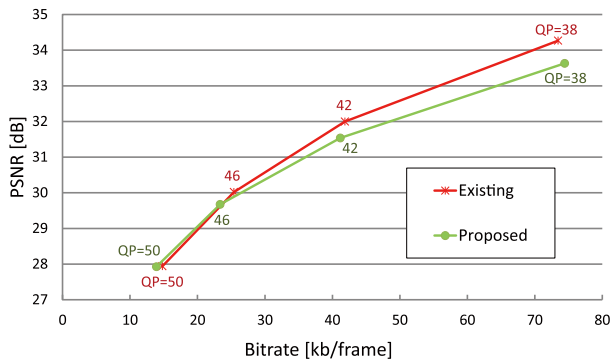
(a) Original image.



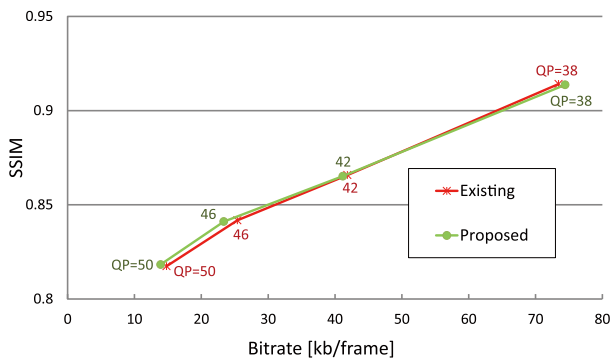
(b) Existing method (QP= 50).



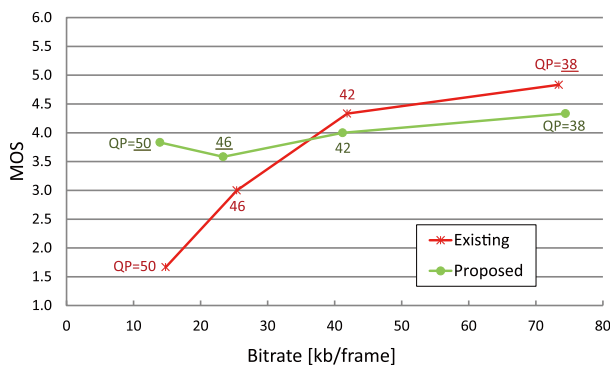
(c) Proposed method (QP= 50).

Fig. 12 Expanded image (Carbon).

(a) PSNR.



(b) SSIM.



(c) Readability.

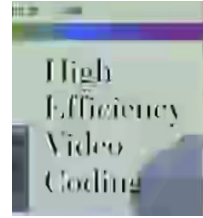
Fig. 13 Result 3 (Book).

shown in Fig. 12.

In the case of Fig. 8 (c), the PSNR and SSIM for the proposed method were almost the same as those of the existing method, whereas the readability was much higher at a low bitrate, as shown in Fig. 13. The text image was heavily deteriorated by the existing method, whereas the readability was preserved by the proposed method, as shown in Fig. 14. It should be noted that the readability for QP = 50 is higher than that for QP = 46. This



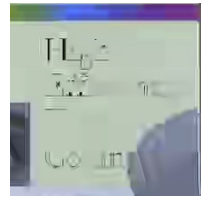
(a) Original image.



(b) Existing method (QP= 50).



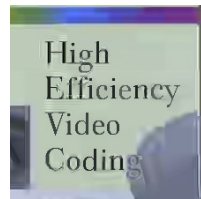
(c) Proposed method (QP= 50).

Fig. 14 Expanded image (Book).

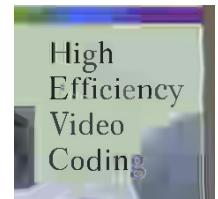
(a) Difference image (QP= 46).



(b) Difference image (QP= 50).



(c) Reconstructed image (QP= 46).

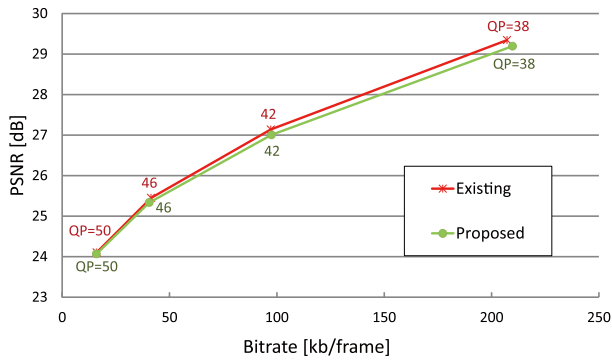


(d) Reconstructed image (QP= 50).

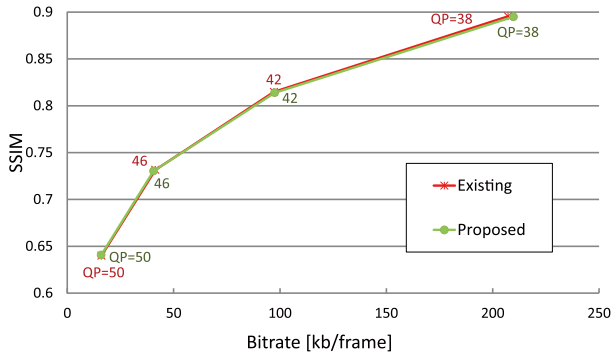
Fig. 15 Comparison between QP = 46 and QP = 50 (Book).**Fig. 16** Character image with a two-tone background (Book).

is because the outlines of the characters remained in the difference image when QP = 46. These outlines disappeared owing to the high compression when QP = 50, and the character images are finely reconstructed. The difference and reconstructed images are shown in Fig. 15. It should also be noted that the PSNR for the proposed method was slightly lower than that of the existing method at a high bitrate, as shown in Fig. 13 (a). In this image, there was a character with a two-tone background, as shown in Fig. 16. The proposed method assumes that the background color is monochromatic, and sometimes the character region could not be detected correctly. To cope with a complex background like this, the introduction of techniques such as image matting may be useful.

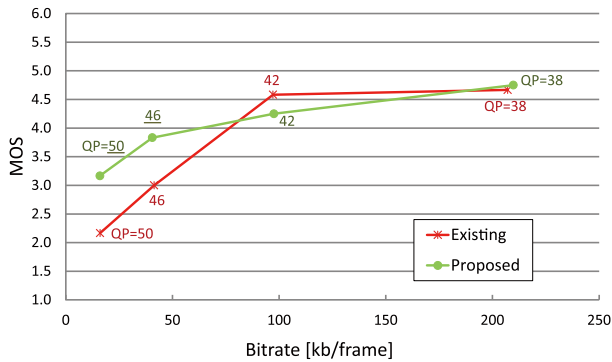
The results for Fig. 8 (d) and (e) are displayed in Fig. 17 and Fig. 18, respectively. The same tendency was observed for these videos. The PSNR and SSIM were almost the same as the exist-



(a) PSNR.



(b) SSIM.



(c) Readability.

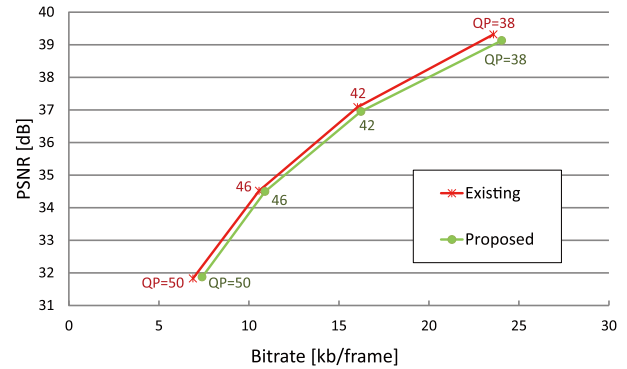
Fig. 17 Result 4 (Card).

ing method, and the readability was higher than that for the existing method at a low bitrate. Even at high bitrates, there was no significant difference between the MOSs. In the case of Fig. 8 (e), the reconstruction of the character images did not work well, as shown in **Fig. 19** (c). However, the readability is much higher than that of the existing method.

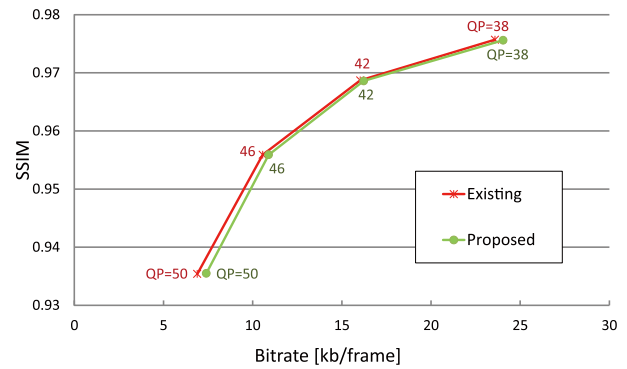
The results for Fig. 8 (f) are shown in **Fig. 20**. The readability of the proposed method was not higher than that of the existing method for every QP value, as shown in Fig. 20 (c). Since the text in this video was relatively large and clear, the video did not deteriorate, even at a very low bitrate, as displayed in **Fig. 21**. In other words, a high readability was also retained by the existing method. The effect of the proposed method is limited in this type of situation.

4. Concluding Remarks

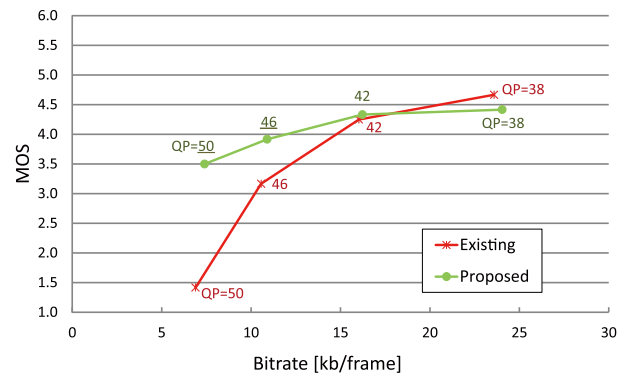
In this paper, we have proposed an efficient coding method with text image generation. From the input video sequences,



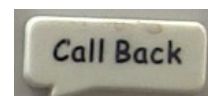
(a) PSNR.



(b) SSIM.



(c) Readability.

Fig. 18 Result 5 (Callback).

(a) Original image.



(b) Existing method (QP= 50).



(c) Proposed method (QP= 50).

Fig. 19 Expanded image (Callback).

eight parameters are estimated for each character for generating images of the text areas. The difference image of the text image and the original image are generated, and the difference image is encoded by the HEVC encoding technique. By combining the encoded image and character parameters, the image including text is reconstructed.

Experiments were carried out with six video sequences. The PSNR, SSIM, and readability were compared for the standard

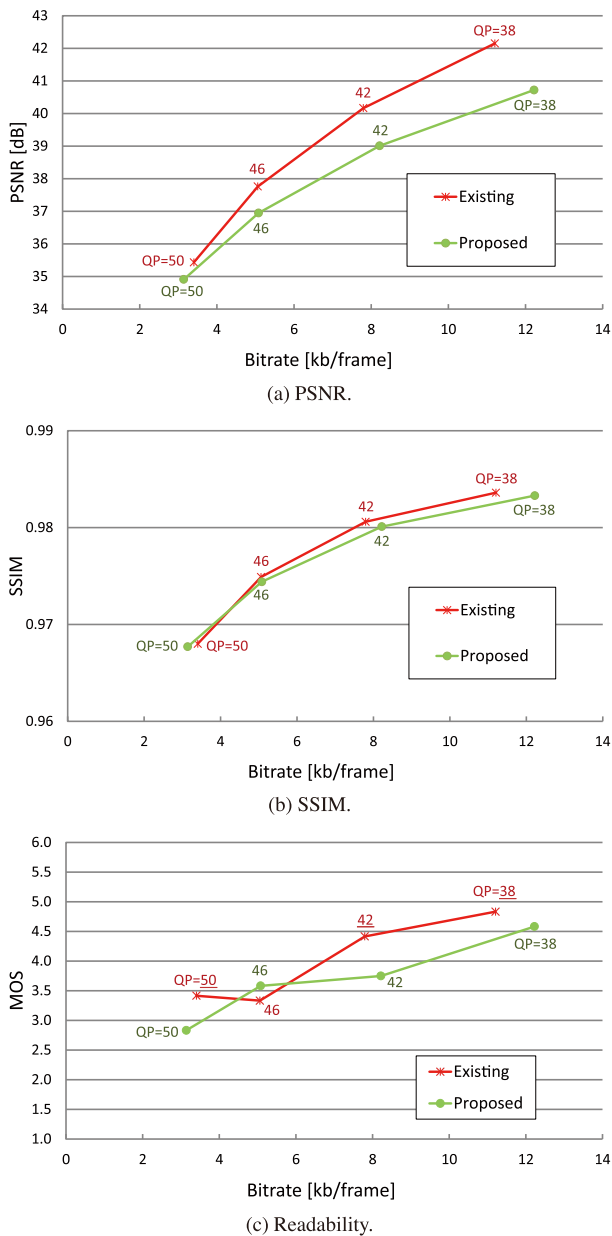


Fig. 20 Result 6 (White wall).

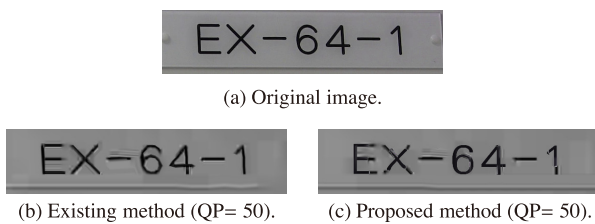


Fig. 21 Expanded image (White wall).

HEVC coding technique and the proposed method. It was verified that the proposed method can improve the readability of text, especially at a very low bitrate, whereas the PSNR and SSIM are almost the same as those of the existing method.

Although the proposed method could improve the readability, a bad impression may result because of the unbalanced positions, sizes, and fonts of the adjacent characters since the proposed method deals with characters independently. It may be effective to treat an entire word or entire sentence simultaneously. In addition,

the correlation between frames is also important. Improvements in the quality by considering these issues are planned for future work.

In the experiment, we only used I-frames. In order to exploit the performance of HEVC, B- and P-frames should also be employed. The introduction of inter frame predictions to the proposed algorithm to utilize these frames is planned for future work. Further, large-scale experimental validation is also important future work.

Acknowledgments This work was partially supported by JST, CREST, and JSPS KAKENHI Grant Numbers 24300066 and 26540077.

References

- [1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2014–2019 White Paper (online), available from http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html (accessed 2015-06-02).
- [2] Wiegand, T., Sullivan, G.J., Bjøntegaard, G. and Luthra, A.: Overview of the H.264/AVC Video Coding Standard, *IEEE Trans. Circuits and Systems for Video Technology*, Vol.13, No.7, pp.560–576 (2003).
- [3] Sullivan, G.J., Ohm, J.-R., Han, W.-J. and Wiegand, T.: Overview of the High Efficiency Video Coding (HEVC) Standard, *IEEE Trans. Circuits and Systems for Video Technology*, Vol.22, No.12, pp.1649–1668 (2012).
- [4] Hu, S., Cohen, R.A., Vetro, A. and Kuo, C.-C.J.: Screen Content Coding for HEVC Using Edge Modes, *Proc. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp.1714–1718 (2013).
- [5] Wang, Z., Yu, Y. and Zhang, D.: Best Neighborhood Matching: An Information Loss Restoration Technique for Block-Based Image Coding System, *IEEE Trans. Image Processing*, Vol.7, No.7, pp.1056–1061 (1998).
- [6] Yang, J., Yin, B., Sun, Y. and Zhang, N.: A Block-Matching Based Intra Frame Prediction for H.264/AVC, *Proc. 2006 IEEE International Conference on Multimedia and Expo*, pp.705–708 (2006).
- [7] Ma, Z., Wang, W., Xu, M. and Yu, H.: Advanced Screen Content Coding Using Color Table and Index Map, *IEEE Trans. Image Processing*, Vol.23, No.10, pp.4399–4412 (2014).
- [8] Puri, A. and Eleftheriadis, A.: MPEG-4: An Object-Based Multimedia Coding Standard Supporting Mobile Applications, *Mobile Networks and Applications*, Vol.3, No.1, pp.5–32, (1998).
- [9] Anh, N.T.N., Yang, W. and Cai, J.: Seam Carving Extension: A Compression Perspective, *Proc. 17th ACM International Conference on Multimedia*, pp.825–828 (2009).
- [10] Deng, C., Lin, W. and Cai, J.: Content-Based Image Compression for Arbitrary-Resolution Display Devices, *IEEE Trans. Multimedia*, Vol.14, No.4, pp.1127–1139 (2012).
- [11] Ndjiki-Nya, P., Hinz, T., Stuber, C. and Wiegand, T.: A Content-Based Video Coding Approach for Rigid and Non-Rigid Textures, *Proc. 2006 IEEE International Conference on Image Processing*, pp.3169–3172 (2006).
- [12] Dumitras, A. and Haskell, B.G.: An Encoder-Decoder Texture Replacement Method with Application to Content-Based Movie Coding, *IEEE Trans. Circuits and Systems for Video Technology*, Vol.14, No.6, pp.825–840 (2004).
- [13] Aziz, H.M., Fiedler, M., Grahn, H. and Lundberg, L.: Compressing Video Based on Region of Interest, *IEEE EuroCon 2013*, pp.1645–1652 (2013).
- [14] Zhang, F. and Bull, D.R.: A Parametric Framework for Video Compression Using Region-Based Texture Models, *IEEE Journal of Selected Topics in Signal Processing*, Vol.5, No.7, pp.1378–1392 (2011).
- [15] Luo, H.: Image-Dependent Shape Coding and Representation, *IEEE Trans. Circuits and Systems for Video Technology*, Vol.15, No.3, pp.345–354 (2005).
- [16] Zhang, H., Zhao, K., Song, Y. and Guo, J.: Text Extraction from Natural Scene Image: A Survey, *Neurocomputing*, Vol.122, pp.310–323 (2013).
- [17] Ye, Q. and Doermann, D.: Text Detection and Recognition in Imagery: A Survey, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.37, No.7, pp.1480–1500 (2015).
- [18] Wang, Z., Bovik, A.C., Sheikh, H.R. and Simoncelli, E.P.: Image Quality Assessment: From Error Visibility to Structural Similarity,

IEEE Trans. Image Processing, Vol.13, No.4, pp.600–612 (2004).

- [19] JCT-VC: HM-12.1 (online), available from https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/ (accessed 2015-06-02).



Yosuke Nozue received B.E. and M.E. degrees from Tohoku University in 2013 and 2015, respectively. He joined Panasonic Corporation in 2015. His research interests include image processing and image coding.



Tomo Miyazaki received B.E. degree from the Department of Informatics, Faculty of Engineering, Yamagata University in 2006. He received M.E. and Ph.D. degrees from the Graduate School of Engineering, Tohoku University in 2008 and 2011, respectively. He joined Hitachi, Ltd. in 2011 and has worked at the Graduate

School of Engineering, Tohoku University from 2013 to 2014 as a researcher. Since 2015, he has been an Assistant Professor. His research interests include pattern recognition and image processing. Dr. Miyazaki is a member of IEICE.



Yoshihiro Sugaya received B.E., M.E., and Doctor of Engineering degrees from Tohoku University, Sendai, Japan in 1995, 1997, and 2002, respectively. He has worked as a Research Associate at the Graduate School of Engineering, Tohoku University from 2000 to 2007, and since 2007, he has been an Assistant Professor.

His research interests include the areas of pattern recognition, parallel processing, distributed computing, parallel architecture, and the smart grid. Dr. Sugaya is a member of IEICE.



Shinichiro Omachi received B.E., M.E., and Doctor of Engineering degrees in Information Engineering from Tohoku University, Japan, in 1988, 1990, and 1993, respectively. He worked as a Research Associate at the Education Center for Information Processing at Tohoku University from 1993 to 1996. Since 1996, he

has been with the Graduate School of Engineering at Tohoku University, where he is currently a Professor. From 2000 to 2001, he was a Visiting Associate Professor at Brown University. His research interests include pattern recognition, computer vision, image processing, image coding, and parallel processing. He received the MIRU Nagao Award in 2007, the IAPR/ICDAR Best Paper Award in 2007, the Best Paper Method Award of the 33rd Annual Conference of the GfKI in 2010, the ICFHR Best Paper Award in 2010, and the IEICE Best Paper Award in 2012. Dr. Omachi is a member of the IEEE and IEICE, among others.