This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2017.2695640, IEEE Transactions on Emerging Topics in Computing

Object-Based Video Coding by Visual Saliency and Temporal Correlation

Kazuya Ogasawara, Tomo Miyazaki, *Member, IEEE*, Yoshihiro Sugaya, *Member, IEEE*, and Shinichiro Omachi, *Senior Member, IEEE*

Abstract—When a disaster occurs, video communication is an effective way to disseminate large quantities of important information. However, video coding standards such as High Efficiency Video Coding (HEVC) compress entire videos, whatever the contents are; at low bit rates, the quality of significant objects deteriorates. In this paper, an object-based video coding method is proposed to address this problem. The proposed method extracts objects on the basis of visual saliency and temporal correlation between frames. Subsequently, we execute preprocessing which degrades the background quality before encoding the video with HEVC. This method can reduce the bit rate while preserving target object quality. Experimental comparison with HEVC demonstrates the superior performance of the proposed method.

Index Terms—object-based coding, object extraction, visual saliency, background blurring

1 Introduction

VIDEO can convey more information than text and still images. Therefore, when a disaster such as the Tohoku Earthquake of March 11, 2011 occurs, video is very useful for quickly and directly spreading important information, such as current conditions and the existence of victims. However, the disadvantage is that the data size of videos is very large. It is problematic to transmit large amounts of video data immediately after a large-scale disaster due to an inevitable increase in communication traffic. Under these circumstances, communication speed and stability may decrease, which can confuse many people.

For that reason, video coding methods that reduce the data size of videos are widely studied. Video coding standards, such as the widely used H.264 and HEVC, have been investigated to support next-generation video service and achieve high compression efficiency. However, in these methods, all of the contents of the video are uniformly compressed. Consequently, important information such as particular objects, which you want to convey, deteriorate when you compress video significantly to reduce its data size. However, in the case of emergencies and many other applications, such as video conferences and surveillance, it is only necessary to transmit regions of interest, rather than the entire video. Therefore, object-based coding scheme, that reduces data size while preserving important information, is effective in such cases.

In this paper, our contribution is to extract visually attractive objects using saliency and temporal correlation in video. In addition, we propose an object-based video coding scheme that degrades only the background before encoding. Then, we demonstrate the effectiveness of the proposed method through experimental results.

The outline of this paper is as follows. In Section 2, related works are introduced. In Section 3, the proposed method is explained in general. In Section 4, we describe the proposed object extraction method in detail. Section 5 presents an object-based coding scheme. Finally, Section 6 shows some experimental results and Section 7 concludes this paper.

2 Related Work

Object-based video coding schemes have been actively investigated. MPEG-4 is a video coding standard for arbitrarily shaped objects. However, since it was standardized in 1999, it may not to preserve important information because the compression efficiency of MPEG-4 is lower than that of current mainstream coding standards. Lan et al. proposed an object-based coding scheme that determines the foreground using a depth map and incorporates the technology of MPEG-4 into HEVC [1]. The disadvantage is that it requires a special video camera to capture the depth map. For videos captured by a stationary camera, there is some research on video surveillance [2] and video conferences [3]. In [4], Ng et al. proposed an object-based coding system using multiple video cameras for dynamic image-based representations. In sum, all of the above methods focus on videos obtained under specific conditions, which is disadvantage.

Many coding schemes have been developed for the purpose of preserving quality in areas that attract human attention. In [5], the foreground is determined on the basis the perceptual characteristic of humans pay attention to fast moving objects. In [6], the facial region is set as the foreground, and parts of the face (e.g., eyes, nose and mouth) are compressed in high quality. In [7], the background is judged whether its texture is liquid, that is not important for viewers and can cause the data size to increase, and a different coding scheme is applied depending on the texture. However, in these coding systems, the

• K. Ogasawara, T. Miyazaki, Y. Sugaya, and S. Omachi are with Graduate School of Engineering, Tohoku University.

xxxx-xxxx/0x/$xx.00 © 200x IEEE Published by the IEEE Computer Society

2168-6750 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2017.2695640, IEEE Transactions on Emerging Topics in Computing

target object is restricted.

There is also research that uses saliency to determine important regions in videos. A coding method that changes the quality parameter by the macroblock based on saliency is investigated in [8]. Because this method [8] uses saliency responses directly to compress videos, it does not separate foreground and background. Hence, the method does not take into account the shape and the contour of the objects. As a preprocessing approach, Cavallaro et al. divides a scene into foreground and background based on a priori information, and then more distortion is imposed to the background in [9]. In [10], Boccignone et al. assumed human faces to be the eye fixation point and then applies a foveation filter to increase the blurring effect along with the distance between the considered pixel and the point. Likewise, Kortum et al. used an eye tracker to record the point-of-gaze and then applies a foveation filter considering the point in [11].

There are two necessary points for more effective object-based coding scheme. First, its coding efficiency is higher than that of HEVC. Second, it does not restrict photographing conditions and targets.

3 Overview of the Proposed Method

We propose an object-based coding system using HEVC. This system extracts visually attractive objects and deteriorates the unimportant background area before HEVC encoding. In this way, it can reduce bit rate while preserving the quality of target objects. A block diagram of the proposed coding system is shown in Fig. 1. The proposed method consists of two steps, object extraction step and background blurring step.

In the object extraction step, we extract visually attractive objects automatically from the input video. We propose an object extraction method that combines GrabCut [12], which is an existing segmentation algorithm, and a saliency map to extract visually attractive objects with high accuracy. We use temporal correlation, which is a characteristic of video, to make the extraction process easy in many frames. In addition, to improve the extraction accuracy, we conduct some processes, such as preprocessing, post-processing, face mask composition, and mask evaluation. This approach aims to extract visually attractive objects, not restricted objects such as fast moving objects, human faces, and so on.

Then, in the background blurring step, we deteriorate only the background considering a human visual characteristic, which is that humans are not conscious of the points far from the point of interest. We change the degree of deterioration depending on the distance between the extracted object and the target background pixel. This approach can reduce unimportant information while maintaining a natural appearance.

Finally, we encode this deteriorated video using HEVC. We can reduce bits in background because spatial and temporal correlations of adjacent background pixels become higher, and accuracy of prediction in HEVC improves by background blurring. Therefore, we can reduce bit rate of a video while preserving the quality of target objects. In other words, we can achieve higher coding efficiency than HEVC.

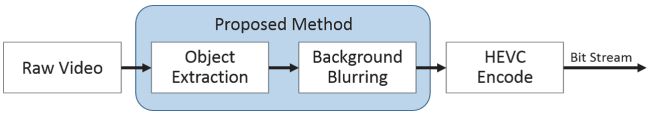


Fig. 1. Block diagram of the proposed object-based coding system.

4 Visually Attractive Object Extraction

This process automatically extracts visually attractive objects. From the input video frames, key frames are chosen at a constant interval. We use an interval of 15 frames in our experiments. In each key frame, we create a mask that shows the target object area using an original segmentation method. This extends GrabCut [12] by introducing visual saliency information. In other frames, we create a mask by referring to the previous mask and calculating the label change only around the object's contour. This improves object extraction by using the previous extraction result. It also reduces processing time by not creating masks for each frame from scratch. Furthermore, in order to improve the accuracy of object extraction for key frames, we compare the created mask with a predicted mask based on the mask of the preceding key frame. If the difference is large, create a mask again in a different setting.

4.1 Key Frame Mask Creation

4.1.1 Pre-Processing

Pre-processing is performed to improve the accuracy of mask creation. First, the size of input frames is reduced to make the subsequent process easier. We reduce frames down to size 720x404 because we use FHD (1920x1080) resolution video in the experiments. Second, smoothing with a Gaussian filter is performed to reduce the high-frequency components, which have a bad influence on segmentation. We empirically set the kernel size to 9 and the standard deviation to 4 in our experiments. Fig. 2 shows a pre-processed image.

4.1.2 Mask Creation

This process creates a mask of the objects using color information and visual saliency. We use the saliency map proposed in [13] for this information. First, we estimate target objects using the saliency map. The saliency map shows the degree of visual attention, and for each pixel i the saliency value $S(i)$ is normalized to the range [0, 1]. In the saliency map, large values represent locations that human will pay attention to. Then, we refine these saliency values referring to [14] and [15]. We suppose that a pixel is in a more attractive area if its saliency value satisfies $S(i) > 0.8$. Since pixels which are closer to the more attractive area should be visually more significant than those which are farther away, we use the refined saliency value, $\hat{S}(i)$, defined as follows:

2168-6750 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2017.2695640, IEEE Transactions on Emerging Topics in Computing

Fig. 2. Pre-processing: (a) Original video frame. The frame size is 1920x1080. (b) The frame after pre-processing. The frame size is 720x404.

Fig. 3. Saliency map: (a) Saliency map derived by [13]. (b) Saliency map refined by (1). The saliency values near the edge of the image become lower than that of (a).

$$\hat{S}(i) = S(i) * (1 - dist_{sal}(i) * 0.75), \tag{1}$$

where $dist_{sal}(i)$ is the Euclidean distance between pixel i and the pixel in attractive area which is the nearest to i. The $dist_{sal}(i)$ is normalized to the range [0, 1]. Fig. 3 shows saliency maps.

The saliency map does not always show uniform saliency across the same object, and sometimes there are pixels with high saliency in background. Therefore, it is difficult to create a mask using only the saliency map. Accordingly, in order to create a highly precise mask, we propose a color and saliency-based segmentation method, which combines the saliency map and GrabCut. GrabCut is a graph-based two-class segmentation method. In the algorithm, the input image is expressed as a weighted graph. Each pixel is represented as a node. In addition, there are two nodes that represent foreground and background, namely Source and Sink. Edges between a pixel and its adjacent pixel is weighted based on color difference between these pixels. Edges between a pixel and Source or Sink are weighted based on color information, such as color likelihood, obtained from foreground and background color GMMs. These color GMMs are generated using foreground and background samples selected by the user. Then, segmentation is conducted by finding the minimum cut to divide the weighted graph into two subgraphs. In GrabCut, a label mask is created based on user interaction to assign sample pixels and generate the color GMMs. The label mask consists of four values: BGD (a background sample pixel), FGD (a foreground sample pixel), PR_BGD (probably a background pixel), and PR_FGD (probably a foreground pixel). The foreground color GMM is generated from color information of the FGD and PR_FGD pixels, and the background color GMM is generated from that of the BGD and PR_BGD

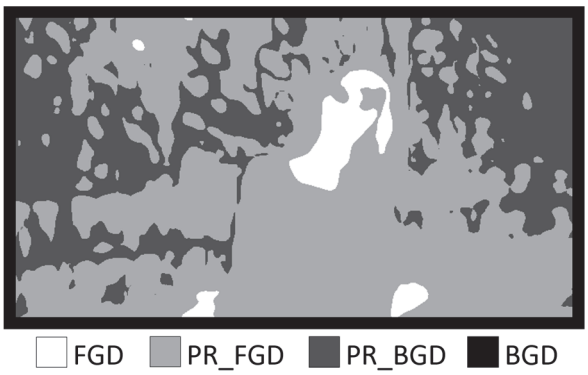


Fig. 4. A label mask created based on the saliency map.

pixels. Each GMM consists of five Gaussian distributions. In addition, the BGD and FGD pixels become background and foreground pixels, respectively. In contrast, the labels of the PR_BGD and PR_FGD pixels are determined by finding the minimum cut of the constructed weighted graph.

In the proposed method, we create the label mask automatically and without user interaction based on the saliency map. First, we hypothesize that humans tend to pay attention to the center of the screen rather than the edge of the screen. Based on this hypothesis, 15 pixels from the border of the image are set as BGD pixels, regardless of its saliency. The other pixels inside the screen are given a label based on the saliency map and a binary threshold T. We obtain the threshold T, with range [0, 1], by applying Otsu's method [16] to the refined saliency map. Fig. 4 shows a label mask. Specifically, pixels with high saliency become FGD pixels, those with middle saliency become PR_FGD pixels, and those with low saliency become PR_BGD pixels as follows:

$$\begin{cases} PR_BGD & \text{if } \hat{S}(i) < T/3 \\ PR_FGD & \text{if } T/3 \le \hat{S}(i) < (T+1)/2 \\ FGD & \text{if } \hat{S}(i) \ge (T+1)/2 \end{cases} \tag{2}$$

Then, an original segmentation method, which extends GrabCut, creates a mask that shows the area of the visually attractive object in the frame. We construct the weighted graph using not only the color likelihood but also the saliency value. The weight of edges between a pixel and Source or Sink are defined as follows:

$$\begin{aligned} Source(i) &= -\log\big(bgdGMM(i)\big) * w_c - \log\left(\frac{1-\hat{S}(i)}{2000}\right) * w_s \\ Sink(i) &= -\log\big(fgdGMM(i)\big) * w_c - \log\left(\frac{\hat{S}(i)}{2000}\right) * w_s \end{aligned} \tag{3}$$

$Source(i)$ and $Sink(i)$ are the weights between pixel i and Source and Sink, respectively. The output color likelihood of the foreground color GMM and background color GMM are represented by $fgdGMM(i)$ and $bgdGMM(i)$, respectively. We empirically set the color weight coefficient w_c to 0.6 and the saliency weight coefficient w_s to 0.4. Finally, we find the minimum cut to divide

2168-6750 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2017.2695640, IEEE Transactions on Emerging Topics in Computing

Fig. 5. A mask created with the original segmentation method.

Fig. 6. The mask after post-processing.

the graph into two subgraphs as with GrabCut. Fig. 5 shows a created mask.

4.1.3 Post-Processing

Three types of post-processings correct mislabeled pixels. First, in order to remove small noise, the created mask is blurred with a median filter, whose kernel size is 9. In the created mask, some of the foreground pixels may be judged as background by mistake. Therefore, we apply the labeling process to the mask and change instances of background labels surrounded by foreground labels to foreground. This process fills holes in the foreground and improves the mask. Likewise, some background pixels may be judged as foreground by mistake. Therefore, we relabel relatively small foreground labels as background; this applies to labels smaller than a quarter of the maximum foreground label size. Fig. 6 shows a post-processed mask.

4.1.4 Face Mask Composition

We consider that human faces draw attention in any video, so faces are important areas. Therefore, we create a face mask that shows a face's area and composes it with the post-processed mask. This process sets the face area as foreground with certainty.

First, we obtain rectangular areas where faces exist by performing face detection with Haar-like features [17]. Next, a label mask for face extraction is created based on this rectangular information. In this face label mask, pixels in a vertically long ellipse in the rectangle are set as

Fig. 7. Face extraction: (a) Face label mask. (b) Face mask. (c) Composed mask.

FGD pixels, the pixels in the remaining region in the rectangle are set as PR_FGD pixels, and pixels in the region outside the rectangle are set as BGD pixels. Finally, a face mask is created using GrabCut and composed with the post-processing mask. Fig. 7 shows a face label mask, a face mask and a composed mask.

This process extracts a face area. However, false recognition is possible in which case some background pixels may be labeled as foreground by mistake. Therefore, we improve the face mask by comparing and evaluating three successive frames. First, we take an exclusive OR of the first face mask (the face mask of the first frame, which is a key frame) and the second face mask (the face mask of the second frame, which is a non-key frame) and erode it. If there are pixels with nonzero values in the result image, the first face mask is completely different from the second face mask. In other words, there is false face detection in either of these two frames. In such a case, the same process is repeated between the second and third face masks to determine which mask had false detection. When the second and third frames have a similar detection result, meaning there are no pixels with nonzero values, there is a high probability that the first face mask was created by mistake. In this case, a process described later corrects the first face mask. First, the difference between the first and second face masks is found by taking exclusive ORs of them and performing erosion/dilation processing. Then, if the area detected as a face in the first frame is larger than that of the second frame, the extra area in the first is removed because it is thought that a non-face area has been detected as face area. In contrast, if the area detected as a face in the first frame is smaller than that in the second, the difference area is composed with the first face mask because it is thought that the face area is not detected in the first frame. Fig. 8 shows an effective example of the face mask.

4.1.5 Mask Evaluation

For more accurate mask creation, we do a comparative evaluation of the mask that takes advantage of temporal correlation, which is a characteristic of video. Video is the aggregation of continuously captured images, so it is rare that the contents vary greatly between frames. Therefore, it is not desirable for the shape of the mask to vary greatly between frames. However, there is a possibility that the shape of the mask will vary greatly when doing object extraction using information about the target frame only.

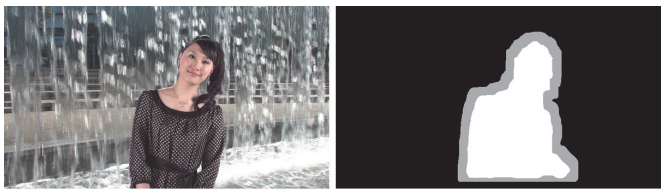
We compare the created mask and the predicted mask, which is predicted from the mask created in the previous

2168-6750 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2017.2695640, IEEE Transactions on Emerging Topics in Computing

(a) (b)

Fig. 8. Effective example of the face mask composition: (a) Before. (b) After.



(a) (b)

Fig. 9. Label mask of a non-key frame: (a) Original non-key frame. (b) Label mask of the non-key frame. The area whose pixels are PR_FGD is the peripheral area of the boundary.

key frame. We use the optical flow to create the predicted mask. First, we calculate the optical flow between continuous key frames based on the method of Farneback [18]. Second, we apply the labeling process to the mask, and calculate the average vector of the optical flow of each label in the foreground area. Then, we create the predicted mask by moving the foreground area for the average vector of each label. The number of pixels where the predicted mask does not overlap with the previously created mask is found by taking an exclusive OR of them. If the number exceeds 5% of the pixels in the frame, we conduct the following process for the mismatched pixels. When the mismatched pixel is a background pixel in the predicted mask, we set that pixel in the label mask as a PR_BGD pixel. Similarly, when it is a foreground pixel in the predicted mask, that pixel in the label mask is set as a PR_FGD pixel. Then, we create a mask again using the renewed label mask.

4.2 Mask Creation in the Non-Key Frame

For the non-key frames, masks are created using temporal correlation. As mentioned in Sec. 4.1.5, it is rare that contents vary greatly between frames. In other words, the shape of the target object does not vary greatly. Therefore, we find changes in labels around the boundary between the foreground and background of the mask created for the previous frame. This approach can make it easier to extract target objects with high accuracy.

First, the pre-processing described in Sec. 4.1.1 is performed. Next, we create a label mask based on the mask of previous frame and conduct GrabCut to create a mask of the target in a non-key frame. In the label mask, pixels outside the peripheral area of the boundary are set as BGD pixels, pixels in the area are set as PR_FGD pixels, and pixels inside the area are set as FGD pixels. In addition, 15 pixels from the border of the image are set as BGD pixels. The peripheral area of the boundary is found by eroding and dilating the boundary. Then, the post-processing described in Sec. 4.1.3 is performed. Finally, the face mask is composed with the mask after correcting for wrongly extracted face area in a way different from that used on the key frame. Assuming that the mask of the key frame was created correctly, foreground pixels found outside the peripheral area of the boundary are the result of false face detection because these pixels are set as BGD in the label mask. Therefore, we correct the labels of the wrongly extracted area by setting all pixels outside the peripheral area of the boundary as background pixels. Fig. 9 shows a label mask of non-key frame.

5 Object-Based Coding Scheme

This process reduces the data size of a video by deteriorating only the background before encoding the video. Deterioration processing is performed as preprocessing before video compression, which makes it easy to add this scheme to a current coding system because there is no necessity to change the codec. In order to deteriorate the background, a Gaussian filter is applied only to the background based on the created mask. It can reduce the bit rate while maintaining a natural appearance by changing the degree of blurring depending on the distance from the extracted objects. In addition, if object extraction is not completely precise, this deterioration mechanism can compensate for that because the area close to target objects is not degraded as much. The overview of this process is shown in Fig. 10.

First, the nearest Euclidean distance between each background pixel and the foreground in the created mask is calculated and normalized to the range [0, 1]. Then, the input frames are reduced to the same size as the created mask, namely 720x404, and its background pixels are blurred with a Gaussian filter, where the standard deviation is determined based on the distance. We define the standard deviation of each background pixel as follows:

$$\sigma_{background}(i) = \begin{cases} dist_{obj}(i) * 7.5 \text{ if } dist_{obj}(i) > \frac{1}{15} \\ 0.5 \text{ if } dist_{obj}(i) \leq \frac{1}{15} \end{cases} \quad (4)$$

where $\sigma_{background}(i)$ is the standard deviation of background pixel i, and $dist_{obj}(i)$ is the smallest Euclidean distance between pixel i and the foreground. The standard deviation of the furthest pixel becomes 7.5, and that of the nearest pixel becomes 0.5. We set the kernel size to 31 regardless of the distance. Finally, the created masks and images with blurred background for each frame are enlarged to the size of the input video, and the enlarged blurred background is composed with the input frame using the enlarged mask. Fig. 11 shows an extracted object image and a degraded frame whose background is blurred.

2168-6750 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2017.2695640, IEEE Transactions on Emerging Topics in Computing

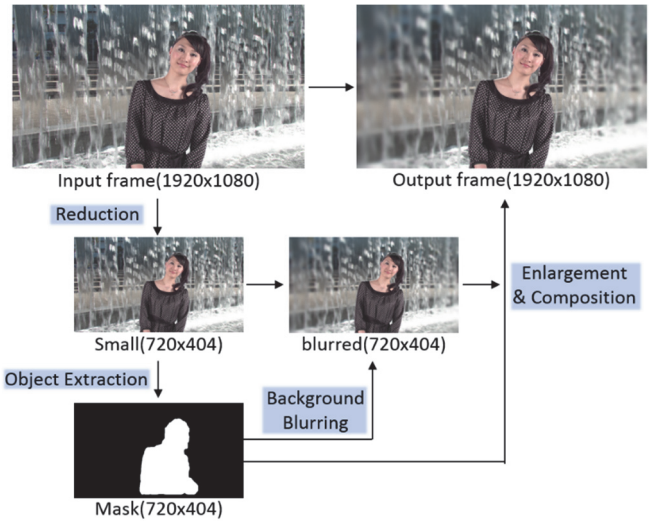


Fig. 10. Overview of background blurring.

Fig. 11. Degraded frame: (a) Extracted object image. (b) Degraded frame.

After the processing described above, we create a video by aggregating the degraded frames, and encode this video with HEVC. Background blurring process increases spatial and temporal correlations of adjacent background pixels, and improves accuracy of prediction in HEVC. In addition, this process does nothing to the foreground and degrades only the background. Therefore, the proposed coding system can reduce bit rate while preserving the visual quality of target objects.

TABLE 1
VIDEO SEQUENCE INFORMATION

Sequences	Cut Out Time	Original Sequences
Truck Train 1	04:00 ~ 06:00	*Truck Train*
Track Train 2	13:00 ~ 15:00	*Track Train*
Fountain 1	30:00 ~ 32:00	*Fountain(dolly)*
Fountain 2	08:00 ~ 10:00	*Fountain(chromakey)*
Fountain 3	16:00 ~ 18:00	*Fountain(chromakey)*

6 EXPERIMENTS

In order to confirm the effectiveness of the proposed method, we conducted a comparative experiment with HEVC which is also used in our method. We used five video sequences reconstituted from the ITE/ARIB Hi-Vision Test Sequence 2nd Edition [19]. Table I shows these videos' sequence information. They are at FHD resolution (1920×1080) and have a frame rate of 30 fps with 60 frames. Example frames from these video sequences are shown in Fig. 12.

First, we introduce results of object extraction and background blurring in Sec. 6.1. Then, in order to confirm coding efficiency of the proposed method, we perform two types of experiments, namely objective evaluation by calculating the rate-distortion curve (Sec. 6.2) and subjective evaluation by measuring mean opinion scores (Sec. 6.3).

6.1 Results of Extraction and Background Blurring

In this subsection, we confirm the accuracy of the proposed object extraction method, and show images whose background is blurred.

First, examples of objects extracted using our extraction method are shown in Fig. 12. We manually label the ground truth for the sequences to evaluate the quality of objects in Sec. 6.2. Fig. 12 shows examples of objects extracted using this ground truth too. In addition, the first frames whose background is blurred are shown in Fig. 13, while that of *Fountain 3* is already shown in Fig. 11 (b).

For almost all video sequences, the proposed extraction method is able to extract objects correctly. However, for *Fountain 1*, a part of the head is misjudged as background in the first frame, and a spray of water is judged as foreground in all frames by mistake. The reason is that the saliency value of a spray of water is high, meaning that the part of the head that is far from it becomes low. Nevertheless, it does not matter much that a part of the background region is judged as foreground by mistake because the goal of the proposed method is to compress videos while preserving the quality of target objects. In addition, as shown in Fig. 13 (c), the part of the head which is judged to be background is not seriously degraded. This is because this part of the head is very close to the foreground label, so that its degree of blurring is low.

2168-6750 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2017.2695640, IEEE Transactions on Emerging Topics in Computing

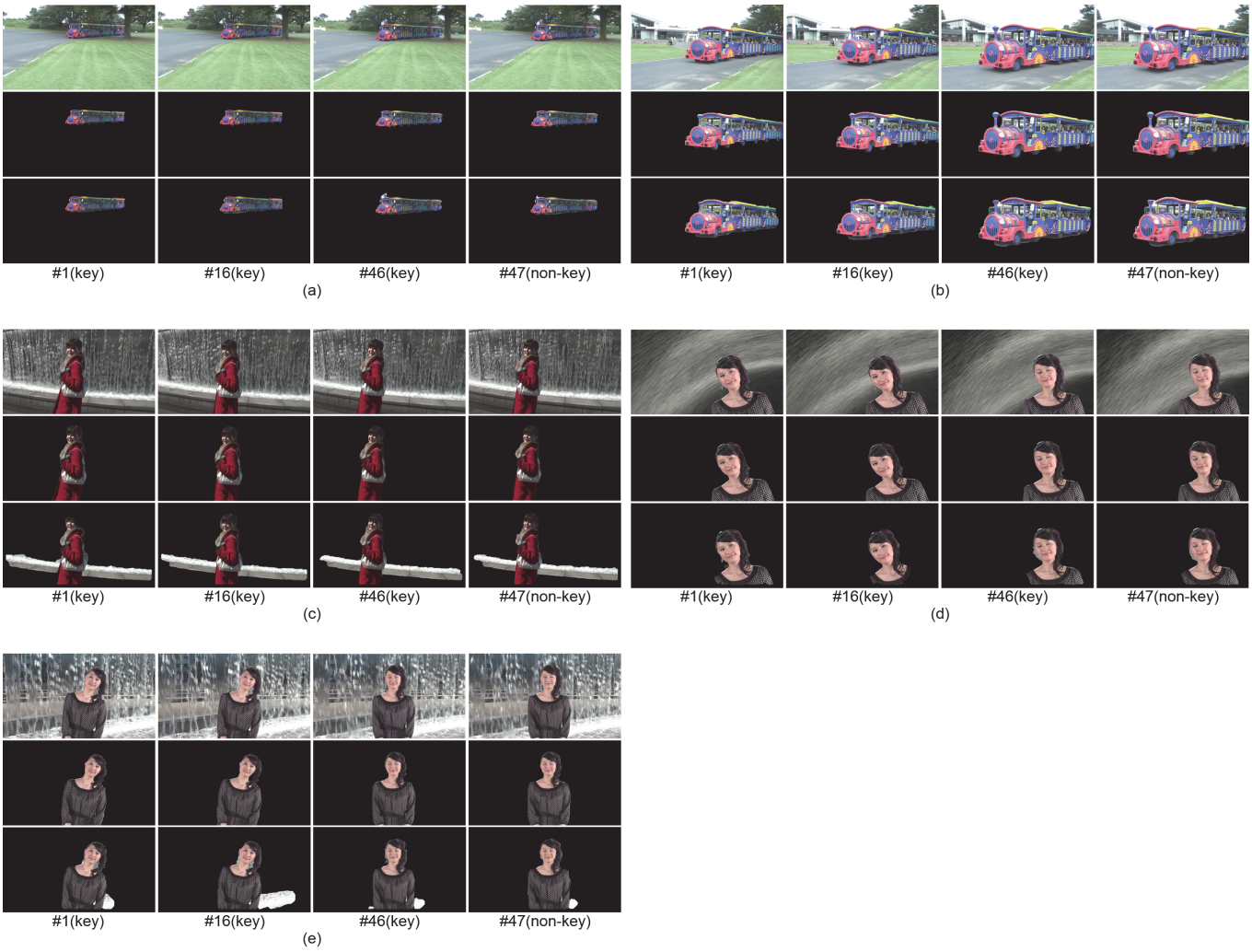


Fig. 12. Examples of extracted object from each frame of different video sequences, namely (a) *Truck Train 1*, (b) *Truck Train 2*, (c) *Fountain 1*, (d) *Fountain 2* and (e) *Fountain 3*. In each subfigure, the original frames are presented in the first row; in the second row the extraction results using ground truth are displayed; extraction results of our method are presented on the bottom of each subfigure.

6.2 Rate-Distortion Performance Evaluation

In this subsection, we evaluated the coding performance using rate-distortion curves, which indicates a relationship between the bit rate and PSNR. Fig. 14 plots the rate-distortion curve of our approach and that of HEVC in whole region and object regions.

In the proposed method, the image quality of the object is good, but that of the background is significantly degraded. Therefore, PSNR on the whole region has low values for almost all of the video sequences at any bit rate. In contrast, video coding standards, such as HEVC utilize rate-distortion optimization to choose the partition manner and the coding mode, so PSNR is optimized. For that reason, PSNR on the whole region was higher for HEVC than for the proposed method.

On the other hand, we confirmed that the PSNRs on the target object were larger for the proposed method than for HEVC at various bit rates for all the video sequences. This is because the proposed method reduces the bits in the background significantly.

This result shows that the proposed method can reduce the bit rate more than HEVC when the image quality of the object is compressed to the same degree between the proposed method and HEVC. However, for *Truck Train 2*, the difference between the PSNR on the object of the proposed method and that of HEVC is small at any bit rate. This is because the background of *Truck Train 2* is not very complicated and does not move greatly, and because the object of *Truck Train 2* is large compared to that of the other video sequences. Therefore, the proposed method does not have a significant effect.

Furthermore, we compare the experimental results of the proposed method and HEVC from the aspect of perceived visual quality. Fig. 15 shows the second reconstructed frame of two video sequences, *Truck Train 1* and *Fountain 3*. It can be seen that the proposed method yields more favorable visual quality with less blurred texture, less block noise and shaper edges in the object area.

2168-6750 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2017.2695640, IEEE Transactions on Emerging Topics in Computing

Fig. 13. First frame whose background is blurred: (a) *Truck Train 1*. (b) *Truck Train 2*. (c) *Fountain 1*. (d) *Fountain 2*.

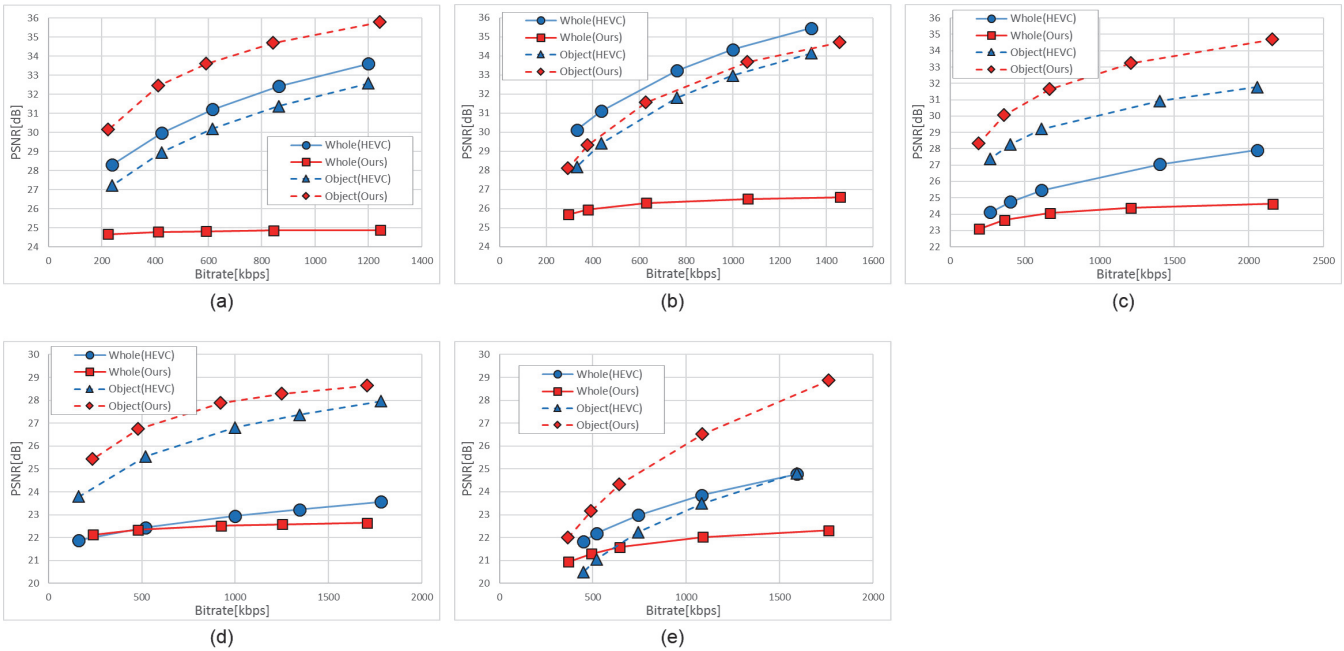


Fig. 14. Rate-distortion curve: (a) *Truck Train 1*. (b) *Truck Train 2*. (c) *Fountain 1*. (d) *Fountain 2*. (e) *Fountain 3*.

6.3 Subjective Quality Evaluation

In this subsection, we evaluate the subjective quality of the object for all the video sequences compressed by the proposed method and HEVC at different bit rates. In this experiment, the uncompressed and compressed video sequences were displayed in a random order. Observers evaluated the quality on a scale of 1 to 10 after viewing. Then, we computed Difference Mean Opinion Scores (DMOS), which indicate the visual difference between the compressed and uncompressed videos. The smaller the DMOS value is the better the subjective quality of the

2168-6750 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2017.2695640, IEEE Transactions on Emerging Topics in Computing

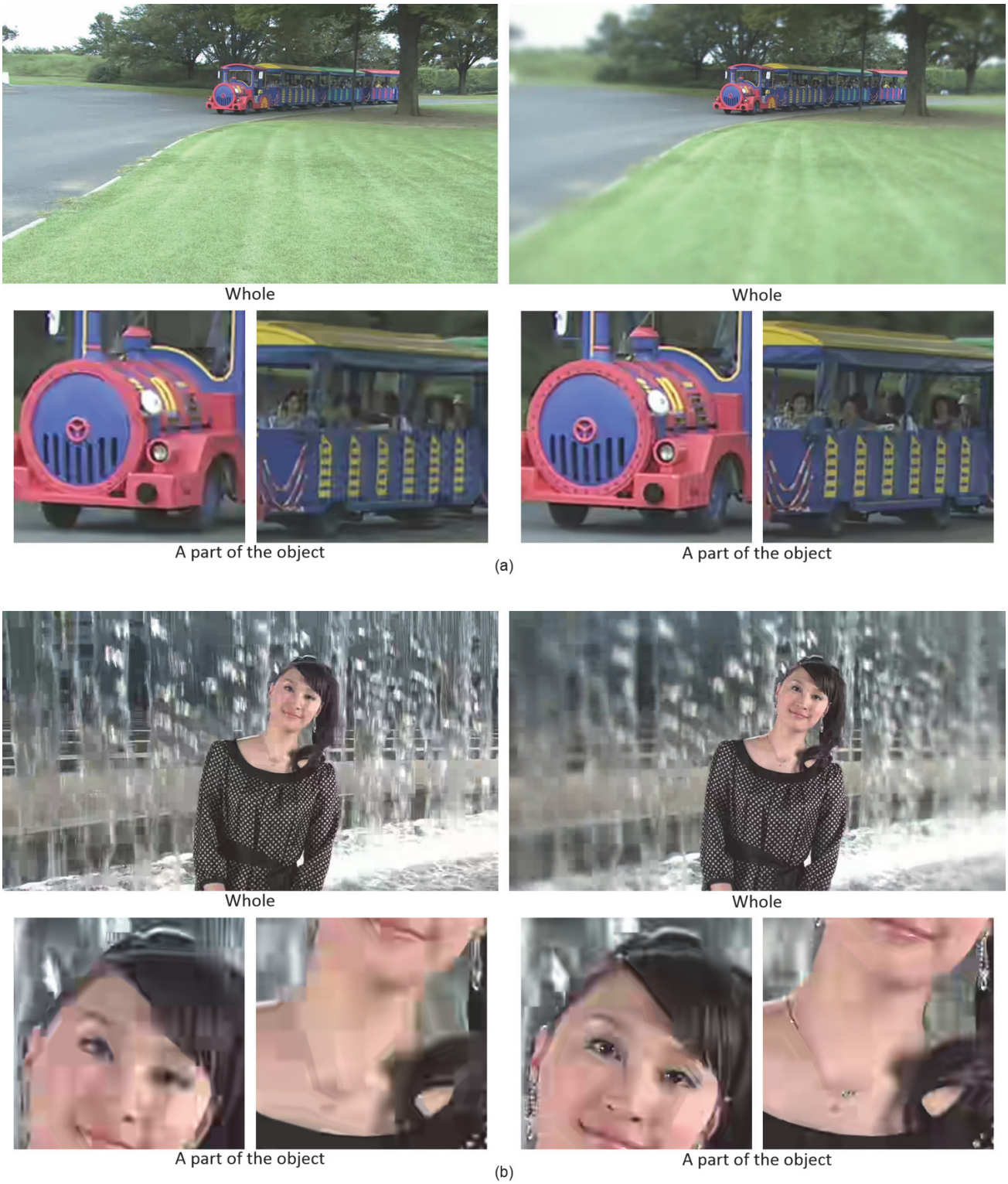


Fig. 15. Visual quality comparison of the reconstructed second frame: (a) *Truck Train 1*. Left images are compressed by HEVC at 238kbps. Right images are compressed by out method at 224kbps. (b) *Fountain 3*. Left images are compressed by HEVC at 520kbps. Right images are compressed by our method at 492kbps.

compressed video sequence. We had 15 participants involved in this experiment.

Fig. 16 shows the average DMOS values. From the figure, we confirm that the average DMOS values of the proposed method were smaller than those of HEVC at any bit rate for all of the video sequences. In other words, our approach can achieve higher subjective quality compared to HEVC. However, for *Truck Train 2*, the differ-

2168-6750 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2017.2695640, IEEE Transactions on Emerging Topics in Computing

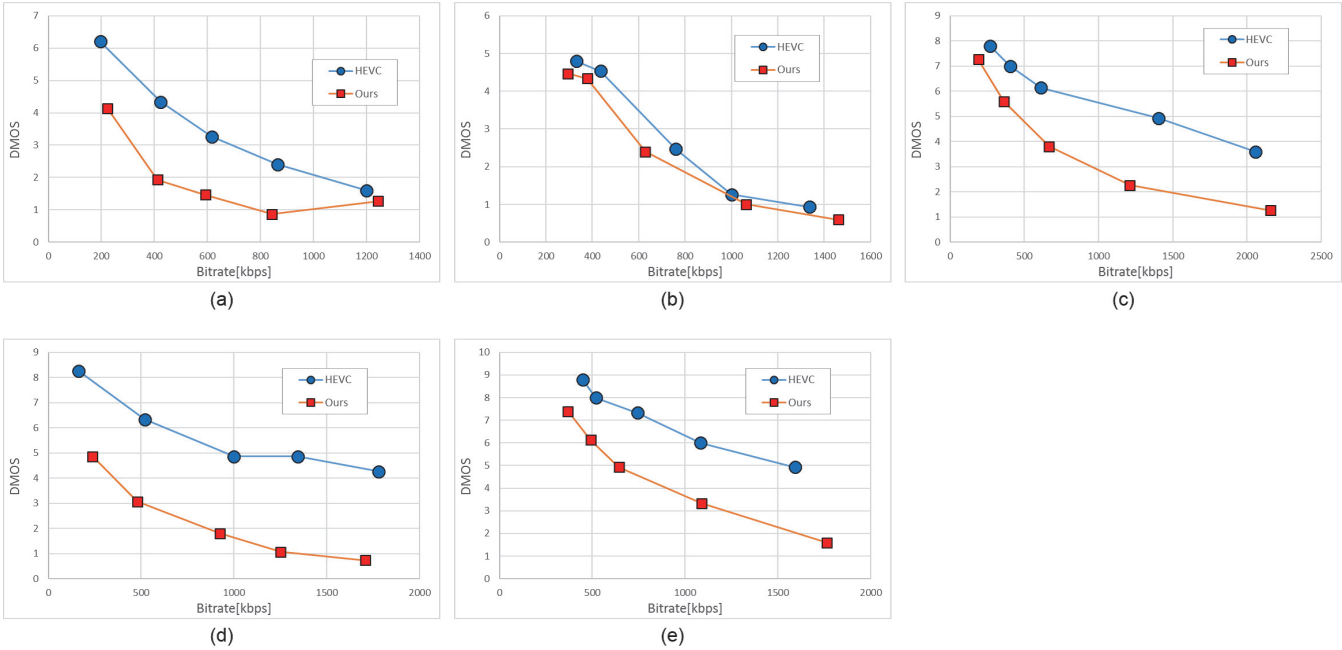


Fig. 16. DMOS value: (a) *Truck Train 1*. (b) *Truck Train 2*. (c) *Fountain 1*. (d) *Fountain 2*. (e) *Fountain 3*.

ence of the DMOS value between our approach and HEVC is small at any bit rates. This is because, as mentioned in Sec. 6.2, our approach was not able to reduce bits much more than HEVC.

7 Conclusion

In this paper, in order to minimize the inevitable increase in communication traffic when a disaster occurs, we proposed an object-based video coding approach for improving the visual quality of target objects. Our method extracts visually attractive objects using visual saliency and temporal correlation, and deteriorates only the background depending on the distance from the object. Then, the video with degraded background is encoded with HEVC. The experimental results show that the proposed method succeeds in both preserving the visual quality of the target objects and reducing the bit rate.

However, there is a possibility of reducing important information greatly if objects are not extracted correctly. The proposed object extraction method in non-key frames may not extract a fast moving object. In addition, if there is a scene change during non-key frames, the proposed extraction method cannot extract objects correctly. Therefore, it is necessary to investigate a more accurate object extraction technique and to introduce a mechanism that can deal with a scene change. In addition, though the proposed extraction approach is designed to be able to extract multiple objects, the video sequences used in the experiments have a single target object. Therefore, we should demonstrate the effectiveness of our method for videos involving multiple objects. Furthermore, the proposed method is not effective for a video that does not contain main objects, such as landscape video. Therefore, it is necessary to introduce a mechanism that judges whether a video contains main objects in order to be effective for any video.

Acknowledgment

This paper is an extended version of the authors' conference paper [20]. The proposed method in this paper is based on HEVC, the latest standard, while the previous method was based on H.264. In addition, face mask was added, and object-based coding scheme was changed.

This work was partially supported by JSPS KAKENHI Grant Numbers 26540077, 16H02841.

References

[1] C. Lan, J. Xu, and F. Wu, "Object-Based Coding for Kinect Depth and Color Videos," Proc. IEEE Conference on Visual Communications and Image Processing, San Diego, 2012, pp. 1-6.

[2] R. V. Babu and A. Makur, "Object-Based Surveillance Video Compression using Foreground Motion Compensation," Proc. ICARCV, Singapore, 2006, pp. 1-6.

[3] Y. Li, X. Tao, and J. Lu, "Hybrid Model-and-Object-Based Teal-Time Conversational Video Coding," Signal Processing: Image Communication, vol. 35, 2015, pp. 9-19.

[4] K.-T. Ng, Q. Wu, S.-C. Chan, and H.-Y. Shum, "Object-Based Coding for Plenoptic Videos," IEEE Trans. Circuits and Syst. Video Technol, vol. 20, no. 4, 2010, pp. 548-562.

[5] M. Bosch, F. Zhu, and E. J. Delp, "Video Coding using Motion Classification," Proc. IEEE International Conference on Image Processing, San Diego, CA, 2008, pp. 1588-1591.

2168-6750 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2017.2695640, IEEE Transactions on Emerging Topics in Computing

[6] M. Xu, X. Deng, S. Li, and Z. Wang, "Region-of-Interest Based Conversational HEVC Coding with Hierarchical Perception Model of Face," IEEE Journal of Selected Topics in Signal Processing, vol. 8, 2014, pp. 475-489.

[7] P. Ndjiki-Nya, T. Hinz, C. Stuber, T. Wiegand, "A Content-Based Video Coding Approach for Rigid and Non-Rigid Textures," Proc. IEEE International Conference on Image Processing, 2006, pp. 3169-3172.

[8] H. Hadizadeh and I. V. Bajić, "Saliency-Aware Video Compression," IEEE Trans. Image Processing, vol. 23, no. 1, 2014, pp. 19-33.

[9] A. Cavallaro, O. Steiger, and T. Ebrahimi, "Semantic Video Analysis for Adaptive Content Delivery and Automatic Description," IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 10, 2005, pp. 1200-1209.

[10] G. Boccignone, A. Marcelli, P. Napoletano, G. D. Fiore, G. Iacovoni, and S. Morsa, "Bayesian Integration of Face and Low-Level Cues for Foveated Video Coding," IEEE Trans. Circuit Syst. Video Technol., vol. 18, no. 12, 2008, pp. 1727-1740.

[11] P. Kortum and W. Geisler, "Implementation of a Foveated Image Coding System for Bandwidth Reduction," Proc. SPIE, vol. 2657, 1996, pp. 350-360.

[12] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: Interactive Foreground Extraction using Iterated Graph Cuts," Proc. ACM SIGGRAPH, 2004, pp. 309-314.

[13] L. Itti, C. Koch, and E. Niebur, "A Model of Saliency-based Visual Attention for Rapid Scene Analysis," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no.11, 1998, pp. 1254-1259.

[14] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-Aware Saliency Detection," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 34, no. 10, 2012, pp. 1915-1926.

[15] W.-T. Li, H.-S. Chang, K.-C. Lien, H.-T. Chang, and Y.-C. Wang, "Exploring Visual and Motion Saliency for Automatic Video Object Extraction," IEEE Trans. Image Processing, vol. 22, no. 7, July 2013, pp. 2600-2610.

[16] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," IEEE Trans. Syst., Man. Cybern., vol. SMC-9, no. 1, Jan. 1979, pp. 62-66.

[17] P. Viola, and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," Proc. CVPR, vol. 1, 2001, pp. 511-518.

[18] G. Farnebäck, "Two-Frame Motion Estimation Based on Polynomial Expansion," Proc. 13th Scandinavian Conference on Image Analysis, 2003, pp. 363-370.

[19] ITE/ARIB Hi-Vision Test Sequence 2nd Edition Reference Manual, 2009.

[20] K. Ogasawara, T. Miyazaki, Y. Sugaya, and S. Omachi, "Object-based Video Coding for Arbitrary Shape by Visual Saliency and Temporal Correlation," Proc. of The Eighth International Conference on Creative Content Technologies, 2016, pp. 13-16.

2168-6750 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.